

AD-A161 154

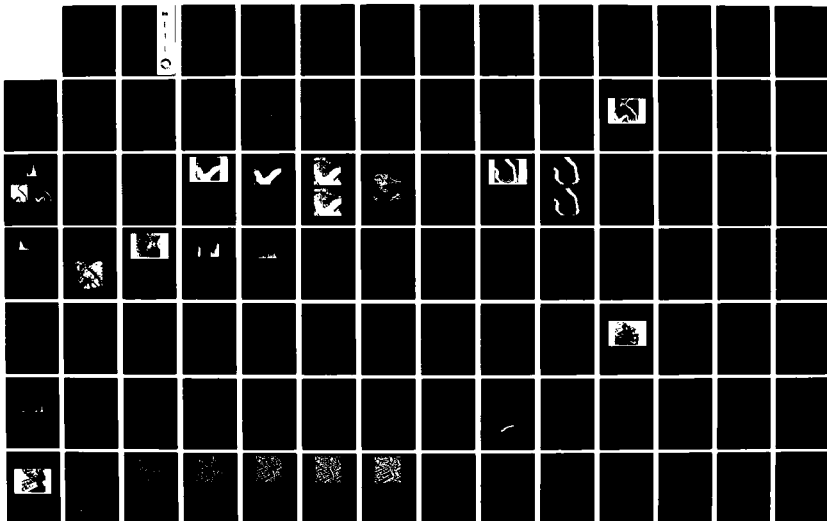
LINEAR FEATURE EXTRACTION FROM RADAR IMAGERY(U)  
ADVANCED INFORMATION AND DECISION SYSTEMS MOUNTAIN VIEW  
CA D T LANTON ET AL. SEP 85 AI/D5-TR-3075-01 ETL-0405  
DACH72-84-C-0014

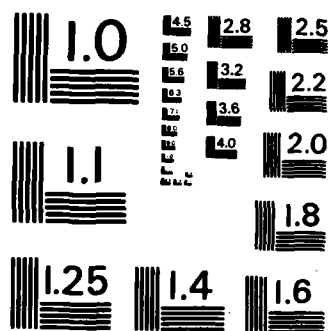
1/2

UNCLASSIFIED

F/G 17/9

NL





ETL-0405

(2)

**AD-A161 154**

# Linear feature extraction from radar imagery

Daryl T. Lawton  
Jay Glicksman  
Gary D. Conner  
Robert J. Drazovich

Advanced Information & Decision Systems  
201 San Antonio Circle, Suite 286  
Mountain View, California 94040

September 1985

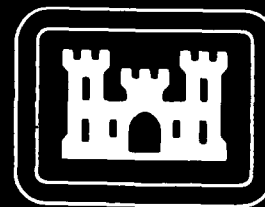
DTIC  
ELECTE  
NOV 13 1985  
S A D

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

**DTIC FILE COPY**

Prepared for  
U.S. ARMY CORPS OF ENGINEERS  
ENGINEER TOPOGRAPHIC LABORATORIES  
FORT BELVOIR, VIRGINIA 22060-5546

85 11 12 158



E

T

L



Destroy this report when no longer needed.  
Do not return it to the originator.

---

The findings in this report are not to be construed as an official  
Department of the Army position unless so designated by other  
authorized documents.

---

The citation in this report of trade names of commercially available  
products does not constitute official endorsement or approval of the  
use of such products.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A161154

## REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY None			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE None				
4. PERFORMING ORGANIZATION REPORT NUMBER(S) TR-3075-01			5. MONITORING ORGANIZATION REPORT NUMBER(S) ETL-0405	
6a. NAME OF PERFORMING ORGANIZATION Advanced Information & Decision Systems		6b. OFFICE SYMBOL (If applicable)		7a. NAME OF MONITORING ORGANIZATION U.S. Army Engineer Topographic Laboratories
6c. ADDRESS (City, State and ZIP Code) 201 San Antonio Circle, Suite 286 Mountain View, CA 94040			7b. ADDRESS (City, State and ZIP Code) Fort Belvoir, VA 22060-5546	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION U.S. Army Engineer Topographic Laboratories		8b. OFFICE SYMBOL (If applicable) ETL-RI-D		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DACW72-84-C-0014
8c. ADDRESS (City, State and ZIP Code) ETL-RI-D Fort Belvoir, VA 22060-5546			10. SOURCE OF FUNDING NOS.	
11. TITLE (Include Security Classification) Linear Feature Extraction from Radar Imagery			PROGRAM ELEMENT NO.	TASK NO.
			PROJECT NO.	WORK UNIT NO.
12. PERSONAL AUTHOR(S) Daryl T. Lawton, Jay Glicksman, Gary D. Conner, Robert J. Drazovich				
13a. TYPE OF REPORT Final Report		13b. TIME COVERED FROM 9/11/84 TO 6/30/85		14. DATE OF REPORT (Yr., Mo., Day) 1985, September
15. PAGE COUNT 148				
16. SUPPLEMENTARY NOTATION				
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB. GR.		
17	09		Artificial Intelligence, Image Understanding,	
06	04		knowledge-based system, expert system, SAR aerial image interpretation	
19. ABSTRACT (Continue on reverse if necessary and identify by block number)				
<p>This report summarizes a feasibility study performed by AI&amp;DS to determine the requirements for the extraction of linear features such as roads, rivers, and environmental region boundaries from SAR aerial imagery. The effort has involved determining effective processes for extracting such features by algorithm surveys, hand processing sample imagery, and algorithm implementation. This work has provided the necessary basis for the implementation of an intelligent, automated system in Phase II of this research. We have designed a general vision system for linear feature extraction which can be generalized to a wide range of SAR (and other sensor) objects, and begun developments of the components of such a system.</p>				
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS <input type="checkbox"/>			21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL P.F. Chen			22b. TELEPHONE NUMBER (Include Area Code) (202) 355-2728	22c. OFFICE SYMBOL ETL-RI-D

Continuation of Block 19:

The motivation for a system-based approach stem from the limited results associated with the undirected application of low level image processing techniques in the extraction of such features and environmental objects. Objects such as roads and rivers are semantic entities whose extraction requires contextual and object-specific knowledge which cannot be easily incorporated into, for example, low level filtering operations.

In our work, we reviewed and implemented several edge, region, and shape extraction routines for application upon SAR aerial imagery. We evaluated their performance and determined which are valuable for integration into a general system. These implemented routines for edge extraction are: The Canny operator, Burt's pyramid, variants of the Hough transform, gradient-based and edge-fragment-based linking. For region extraction: 1D feature histogram-based segmentation, Burt's Hierarchical Discrete Correlation, object based texture classification over image sub-areas, Kohler's algorithm, and plurality updating. For shape characterization: recursive line fitting, chamfer-based medial axis transform, and basic shape measures.

We also designed and partially implemented the Image Structure Data Base (ISDB). This is a basic system component for representing processing results and extracted image structures. We considered a variety of techniques for representing the properties of environmental objects such as roads and rivers in SAR imagery. We have organized the SAR object knowledge into a network of feature attributes and programmed finders.

We have used the components from the ISDB and implemented image processing routines to evaluate several processing scenarios for the extraction of roads, rivers, and region boundaries. This has demonstrated a capability for extracting roads, rivers and region boundaries from SAR imagery using automated processing techniques (selected in an interactive fashion).

Accession For	
NHS	<input checked="" type="checkbox"/>
DHS	<input type="checkbox"/>
U.S.	<input type="checkbox"/>
J. Edgar Hoover	
FBI	
Washington, D.C.	
A-1	

## **PREFACE**

**This report was prepared under contract DACW72-84-C-0014 for the U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia, by Advanced Information & Decision Systems, Mountain View, California. The Contracting Officer's Representative was Dr. Pi-Fuay Chen.**

## TABLE OF CONTENTS

	Page
<b>1. INTRODUCTION</b>	<b>1-1</b>
1.1 EXECUTIVE SUMMARY	1-1
1.2 PROBLEM OVERVIEW	1-4
1.3 REPORT OUTLINE	1-5
<b>2. A VISION SYSTEM FOR SAR IMAGE FEATURE INTERPRETATION</b>	<b>2-1</b>
2.1 MOTIVATION	2-1
2.2 SYSTEM OVERVIEW	2-1
2.2.1 Image Structure Data Base	2-3
2.2.2 Hypothesis/Task Data Base and Manager	2-4
2.2.3 Segmentation Knowledge Source	2-5
2.2.4 SAR Object Knowledge	2-6
2.2.5 System Controller	2-7
2.3 PROCESSING SCENARIO	2-7
<b>3. IMAGE STRUCTURE DATA BASE</b>	<b>3-1</b>
3.1 OBJECTS	3-1
3.1.1 Images	3-2
3.1.2 Image Structures	3-3
3.1.3 Relations	3-9
3.1.4 Non-Image Objects	3-11
3.2 PROCESSING RELATIONSHIP STRUCTURE	3-12
3.3 QUERIES	3-14
<b>4. SEGMENTATION KNOWLEDGE SOURCE</b>	<b>4-1</b>
4.1 IMAGE SEGMENTATION	4-1
4.2 EDGE EXTRACTION PROCEDURES	4-2
4.2.1 Zero-Crossing Extraction	4-3
4.2.2 Canny	4-4
4.2.3 Burt	4-15
4.2.4 Hough Transform	4-21
4.2.5 Gradient Based Edge Linking	4-26
4.2.6 Segment Based Edge Linking	4-26
4.2.7 Nevatia/Babu	4-30
4.2.8 Others	4-39

<b>4.3 REGION SEGMENTATION</b>	<b>4-39</b>
4.3.1 Histogram-Based Segmentation	4-40
4.3.2 Plurality Updating	4-40
4.3.3 Texture Segmentation	4-40
4.3.4 Kohler	4-41
<b>4.4 SHAPE EXTRACTION PROCEDURES</b>	<b>4-45</b>
4.4.1 Extraction of Significant Curvature Points	4-45
4.4.2 Chamfer-Based Shape Descriptions	4-46
4.4.3 Basic Shape Statistics	4-51
<b>4.5 SEGMENTATION RULES</b>	<b>4-51</b>
<b>5. SAR OBJECT KNOWLEDGE REPRESENTATION</b>	<b>5-1</b>
<b>6. SUMMARY AND FUTURE PLANS</b>	<b>6-1</b>
<b>7. BIBLIOGRAPHIES</b>	<b>7-1</b>

## LIST OF FIGURES

	Page
Figure 2-1: System Design	2-2
Figure 2-2: Example SAR Image: ETL6	2-8
Figure 2-3: Contours at Different Spatial Frequencies	2-10
Figure 2-4: Linear Segment Approximations	2-11
Figure 2-5: Intensity Histogram and Extracted Image Areas	2-12
Figure 2-6: Long Connected Contours	2-13
Figure 2-7: Image and Extracted Edges	2-15
Figure 2-8: Histogram Extracted River Superimposed	2-16
Figure 2-9: Selected Edge and Attributes	2-16
Figure 2-10: Selected Edge and Corresponding Binary Image	2-17
Figure 2-11: Extended Potential River Regions	2-17
Figure 2-12: Boundary Constraints on River Tracking Procedure	2-18
Figure 2-13: Image and Extracted Edges	2-20
Figure 2-14: Overlaid River	2-21
Figure 2-15: Long Linear Segments	2-21
Figure 2-16: Edges in River Area	2-22
Figure 2-17: Long Edges in River	2-22
Figure 2-18: Long Linear Segments	2-23
Figure 2-19: Long Linear Segments Positioned in River Region	2-23
Figure 2-20: Extracted Edges	2-25
Figure 2-21: Short Linear Subsegments	2-25
Figure 2-22: Average Intensity Histogram for Edge Segments	2-26
Figure 2-23: Selected Edges Corresponding to Major Cluster	2-26
Figure 2-24: Other Edges	2-27
Figure 2-25: Thresholded Density Plot of Edges from Peak	2-27
Figure 2-26: Image and Extracted Edges	2-28
Figure 2-27: Orientation Histogram	2-29
Figure 2-28: Edges Mapped from Major Cluster	2-29
Figure 2-29: Orientation Histogram from Selected Edges	2-30
Figure 2-30: Edges from Orthogonal Peaks	2-30
Figure 2-31: Edges from Orthogonal Peaks (cont'd)	2-31
Figure 2-32: Extracted Edges	2-31
Figure 2-33: Threshold Linear Approximations	2-32
Figure 2-34: Linked Edge Fragments	2-32
Figure 2-35: Linked Edge Fragments (cont'd)	2-33
Figure 2-36: Linked Edge Fragments Superimposed on Extracted Edges	2-33
Figure 2-37: Linked Edge Fragments Superimposed on Extracted Edges (cont'd)	2-34
Figure 3-1: Object-Label Grid	3-7
Figure 3-2: Field Image: (ETL36)	3-15
Figure 3-3: Edge Outputs	3-16
Figure 3-4: Thinned and Labeled Edges	3-17
Figure 3-5: Long Linear Subsegments	3-18
Figure 3-6: Orientation Histogram over Long Linear Subsegments	3-19
Figure 3-7: Extracted Peaks Mapped onto Corresponding Image Structures	3-20

	Page
Figure 3-8: Processing Relationship Structure	3-21
Figure 3-9: Extracted Curves	3-26
Figure 3-10: Selected Curves Based Upon Length and Average Contrast	3-26
Figure 3-11: Selected Curve	3-27
Figure 3-12: Mask Surrounding Curve	3-27
Figure 3-13: Curves Intersecting Mask	3-28
Figure 3-14: Curves with Dominant Intersections with Mask	3-28
Figure 4-1: Selected Field Image: (ETL36 Sub-Image)	4-5
Figure 4-2: Extracted Zero-Crossing Regions and Contours	4-6
Figure 4-3: Extracted Zero-Crossing Regions and Contours (cont'd)	4-7
Figure 4-4: Extracted Zero-Crossing Regions and Contours (cont'd)	4-8
Figure 4-5: Extracted Zero-Crossing Regions and Contours (cont'd)	4-9
Figure 4-6: Extracted Zero-Crossing Regions and Contours (cont'd)	4-10
Figure 4-7: Extracted Zero-Crossing Regions and Contours (cont'd)	4-11
Figure 4-8: Thresholded Zero-Crossing Contours	4-12
Figure 4-9: Thresholded Zero-Crossing Contours (cont'd)	4-13
Figure 4-10: Superimposed Zero-Crossing Contours	4-14
Figure 4-11: Gradient Calculation	4-16
Figure 4-12: Gradient Interpolation	4-17
Figure 4-13: Canny Edges	4-18
Figure 4-14: Canny Edges (cont'd)	4-19
Figure 4-15: Canny Edges (cont'd)	4-20
Figure 4-16: Zero-Crossing Regions from the Laplacian Pyramid	4-22
Figure 4-17: Zero-Crossing Regions from the Laplacian Pyramid (cont'd)	4-23
Figure 4-18: Zero-Crossing Regions from the Laplacian Pyramid (cont'd)	4-24
Figure 4-19: Thresholded Contours	4-25
Figure 4-20: Hough Transform Parameterization	4-27
Figure 4-21: Orientation Histogram	4-28
Figure 4-22: Selected Image Points Corresponding to Peak in Orientation Histogram	4-28
Figure 4-23: Edge Segments	4-31
Figure 4-24: Linked Edge Segments	4-32
Figure 4-25: Linked Edges Segments (cont'd)	4-33
Figure 4-26: Linked Edge with Respect to Extracted Segments	4-34
Figure 4-27: Successor Neighborhood	4-35
Figure 4-28: Field Image: (ETL17)	4-36
Figure 4-29: Nevatia-Babu Edge Outputs	4-37
Figure 4-30: Nevatia-Babu Edge Outputs (cont'd)	4-38
Figure 4-31: Selected Edge Fragments	4-42
Figure 4-32: HDC - Level 1	4-42
Figure 4-33: HDC - Level 2	4-43
Figure 4-34: HDC - Level 3	4-43
Figure 4-35: Region Based Forest Texture Invariants	4-44
Figure 4-36: Extracted Edges	4-47
Figure 4-37: Edges Selected on Average Contrast	4-47
Figure 4-38: Recursive Line Fit Linear Approximations	4-48
Figure 4-39: Interesting Points Positioned with Respect to the Original Contour	4-49
Figure 4-40: Linear Segments	4-49
Figure 4-41: Interpolated Smoothed Contour	4-50
Figure 4-42: Orientation Differences Along Contour	4-50
Figure 4-43: River Segment	4-52

	Page
Figure 4-44: Extracted Boundaries	4-52
Figure 4-45: Contour of Image Chamfer	4-52
Figure 4-46: Label Distance Chamfer	4-53
Figure 4-47: Chamfer-Label Boundary from River Segments	4-54
Figure 4-48: Multi-Resolution Shape Skeletons from Chamfer-Labeling	4-55
Figure 4-49: Example Segmentation Rules	4-57
Figure 4-49: Example Segmentation Rules (cont'd)	4-58
Figure 4-49: Example Segmentation Rules (cont'd)	4-59
Figure 5-1: Object Subnetwork	5-3
Figure 5-2: Object Subnetwork	5-4

## 1. INTRODUCTION

This document is the final report on a research effort undertaken by Advanced Information & Decision Systems (AI&DS) as a partial fulfillment of U.S. Army contract #DACW72-84-C-0014 for the U.S. Army Engineer Topographic Laboratories. The effort is focused on developing automated techniques for extracting linear features (e.g., roads, rivers, boundaries between regions, etc.) from aerial scenes imaged by a synthetic aperture radar (SAR) imaging sensor. This project is a Phase I effort in the government's Small Business Innovative Research (SBIR) program. It is directed toward analyzing the feasibility of automated linear feature extraction techniques and developing a system concept that can be prototyped as part of a follow-on Phase II effort.

### 1.1 EXECUTIVE SUMMARY

An increasingly important task facing numerous government and DoD-agencies is the ability to automatically analyze aerial images. The applications include a variety of intelligence and surveillance tasks that use a variety of image sensors. This report summarizes a feasibility study performed by AI&DS to determine the requirements for the automated extraction of linear features such as roads, rivers, and environmental region boundaries from SAR aerial imagery. The effort has involved determining effective processes for extracting such features by analyzing and testing a variety of algorithms and techniques. This work has provided the necessary basis for the implementation of an intelligent, automated system in Phase II of this research. A general vision system for linear feature extraction has been designed and development of the components of the system have been initiated. The design provides a general framework that can be extended to the automated analysis of a wide range of other SAR (and other sensor) objects.

The primary motivation for such a system-based approach stems from the limited results associated with the undirected application of low level image processing techniques in the extraction of such features and environmental objects. Objects such as roads and rivers are semantic entities whose extraction requires contextual and object-specific knowledge which cannot be easily incorporated

into, for example, low level filtering operations. Our work has made it clear that a general and expandable system will have to incorporate processing which reflects the actual reasoning involved in expert SAR image interpretation.

The major accomplishments of this study have been to:

- 1) Develop a general system architecture for processing aerial SAR imagery. The design is focused around two central data bases that maintain image structures and hypotheses. These data bases are used by the system's various processing algorithms to review previous analyses and to store new results and are used by control algorithms to intelligently and opportunistically select image analysis activities.
- 2) Review and implement several edge, region, and shape extraction routines for application upon SAR aerial imagery. Their performance was evaluated and their value for integration into a general system was analyzed. This work is summarized in Section 4 of this report. These implemented routines for edge extraction are:

- the Canny operator
- the Burt's pyramid
- variants of the Hough transform
- gradient-based linking
- edge-fragment-based linking

For region extraction:

- 1D feature histogram-based segmentation
- Burt's Hierarchical Discrete Correlation
- object based texture classification over image sub-areas
- Kohler's algorithm
- plurality updating

For shape characterization:

- recursive line fitting
- chamfer-based medial axis transform
- basic shape measures
- extensions to shape extraction using chamfering
- local, iterative process to determine points of significant curvature

- 3) Design and partially implement the Image Structure Data Base (ISDB). This is a basic system component for representing processing results and extracted image structures. This work is summarized in Section 3 of this report.
- 4) Consider a variety of techniques for representing the properties of environmental objects such as roads and rivers in SAR imagery. The SAR object knowledge was organized into a network of feature attributes and programmed finders. Automatic generation of these from world knowledge and a priori models was considered. This work is summarized in Section 5.
- 5) Use the components from the ISDB and implemented image processing routines, to evaluate several processing scenarios for the extraction of roads, rivers, and region boundaries. This has demonstrated a capability for extracting roads, rivers and region boundaries from SAR imagery using automated processing techniques (selected in an interactive fashion). This work is described throughout this report.
- 6) Design segmentation and bottom-up processing in a modular, rule-based form to allow for intelligent control based upon strategies and object models. This work is summarized in Section 4.
- 7) Obtain a better understanding of the nature of SAR aerial imagery and its requirements for interpretation.
- 8) Study relevant work on hypothesis management and evidential reasoning.

- 9) Gain considerable experience with LISP machines and ZETA-LISP for implementing image processing routines, semi-autonomous vision systems, and user interfaces.

## 1.2 PROBLEM OVERVIEW

Imaging radar sensors provide all-weather, cloud penetration capability for a variety of applications. Technical capabilities now allow enormous volumes of such imagery to be automatically produced in relatively short periods of time. However, the current methods for analysis and interpretation of radar imagery largely consist of manual examination by human experts. As the quantity of imagery expands, the requirements for timely and efficient feature classification and the scarcity of radar image interpreters point to the need for an automated system for feature detection and classification.

Linear features such as roads, rivers, bridges, and railroads are major landmarks in such imagery and extracting and analyzing such features are a prerequisite for most analysis applications. Traditional linear feature extraction techniques (edge detection and region segmentation) tend to perform adequately for low noise, high resolution visible imagery, and in the generation of preprocessed results for evaluation by a human. However, the relatively poor quality and the complexity of the observed scenes in radar imagery make these feature extraction techniques less effective.

Hence, the ability to automatically detect and analyze linear features has major payoffs for numerous applications. Technology to provide such an automated capability is also emerging from the fields of image understanding (IU) and artificial intelligence (AI). Such a system can incorporate knowledge about the scene and use context (from the image or external sources such as digital terrain maps or terrain object models) to intelligently guide and interpret the exploitation process. It can also be organized to reflect the actual interpretation strategies employed by analysts for completely automatic processing or as an intelligent, interactive processing aid.

### **1.3 REPORT OUTLINE**

Section 2 contains an overview of the system architecture briefly describes each major component, and presents an example processing scenario.

Section 3 contains a description of the Image Structure Data Base (ISDB). The ISDB provides the system's basic representation of system processing and results. The type of objects that the ISDB supports, its relation to other system components, the format of queries over this data base, and how the ISDB is implemented using flavors in ZETA-LISP are described.

Section 4 contains summaries and results from the different segmentation and shape description procedures implemented. Segmentation rules which allow these routines to be applied in a task directed manner are also described.

Section 5 describes the representations of world objects and their appearance in SAR imagery. Two basic types are presented: Feature Vectors and Programmed Finders. The implications of these representations for the system hypothesis formation and system control activities are discussed.

Section 6 contains recommendations for future work.

Section 7 contains the bibliography.

## **2. A VISION SYSTEM FOR SAR IMAGE FEATURE INTERPRETATION**

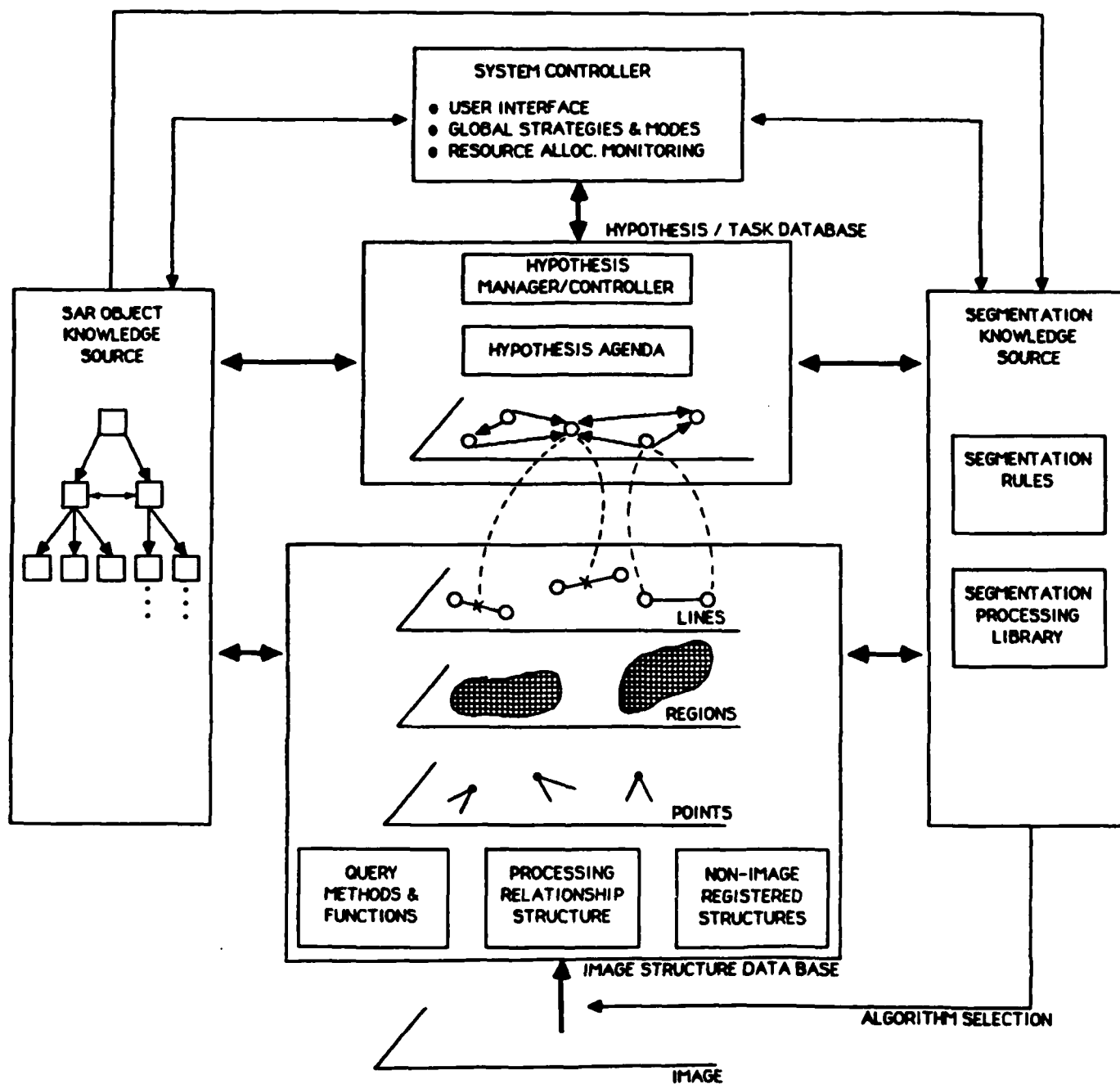
### **2.1 MOTIVATION**

We have two basic motivations for developing a general vision system for SAR imagery interpretation. First, undirected application of lower level image processing techniques will not reliably extract semantically defined world objects like roads, rivers, and bridges. An explicit model of these objects is necessary to direct the application of segmentation procedures and to interpret their results. This requires a system which can represent the properties of world objects to infer their appearance in imagery and which can also apply segmentation knowledge in a flexible, context directed fashion.

Our other major motivation is to develop a general and extendible workstation for SAR image interpretation as the basis of our system development. This workstation will support a wide range of tasks: the interactive exploration of imagery; the development and application of image processing operations; and editing the object representations and processing rules used in the autonomous system.

### **2.2 SYSTEM OVERVIEW**

The general system architecture is shown in Figure 2-1. It consists of two core data bases: the Image Structure Data Base (ISDB) and the Hypothesis/Task Data Base (HTDB). Associated with these data bases are controllers and user interfaces for investigating their contents and status. Surrounding them are the three different system components which access and update the data bases: The Segmentation Knowledge Source, the SAR Object Knowledge Source, and the System Controller. In general, the interpretation process consists of the application of rules and object format descriptions to organize entities in the core system data bases into verifiable hypotheses that correspond to objects and significant image structures.



**Figure 2-1: System Design**

### **2.2.1 Image Structure Data Base**

The Image Structure Data Base (ISDB) represents image processing results and relationships. It consists of several things: images; image registered objects, such as curves, regions, and points; non-image registered objects such as histograms, tables, and networks; the Processing Relationship Structure (PRS) which keeps track of the derivation of objects in the ISDB; and a library of functions and methods for making queries over the ISDB and which form a basic vocabulary for the actions of the other system components.

Entities are dynamically added to the ISDB during the interpretation process. This often involves the results from procedures applied to selected image areas, as in using high frequency edge operators at selected locations during fine edge tracking. This conditional application of image processing routines is indicated by the arrow from the Segmentation Knowledge Source to the arrow from the image to the ISDB. The ISDB also represents the results of several different types of processing for edge and region extraction. Associated with each object is the type of process that extracted it and the relevant parameters. The data base supports the results of image processing at multiple levels of spatial resolution in pyramid data structures or results from operators of different widths.

Interactions with the ISDB take the form of queries for detecting particular image events or relations. These queries are interpreted into the primitive attributes and relations used in the data base and are implemented in a library of functions and methods associated with the ISDB. For example, finding roads and shadowed embankments can involve extracting all long lines that are near each other, have similar orientations and are adjacent to the same set of dark regions. Note that it is important to consider, in the interpretation of queries, what attributes such as LONG and PARALLEL map onto with respect to particular parameter ranges for attributes of structures in the ISDB. There are also significant efficiency considerations with respect to the order in which to extract things (Since there may be fewer long lines than there are dark regions).

Results from queries to the ISDB can be displayed graphically and form the basis of a user interface. Section 3 greatly expands the discussion of the ISDB.

### **2.2.2 Hypothesis/Task Data Base and Manager**

The recognition of objects and more complicated image structures involves grouping operations over the entities in the ISDB and the hypothesis/task data base (HTDB). These operations are specified by Segmentation Rules and by the format of object descriptions in the representations of generic SAR objects. For example, one simple segmentation rule joins lines of similar orientations with nearly adjacent endpoints together. At an object specific level, a road network is the grouping of extracted road segments that are connected together. The basic results of system processing is the set of hypotheses in the HTDB.

The system's interpretations of how image features are grouped or analyzed are represented by hypotheses. Each hypothesis is represented in a common fashion by a symbol and a set of properties that describe the hypothesis. When a hypothesis is generated by the system, it is said to be "instantiated" (an instance of the hypothesis has been found). The properties of the hypotheses include information about the type of hypothesis (e.g., these line segments form a road network, this region is a river, etc.), pointers to the objects being grouped, the strength or certainty of the system's belief in the hypothesis, and the specific information about the objects grouped or analyzed by the hypothesis.

The HTDB consists of an agenda that orders the current hypotheses in terms of the importance of their verification. The order of hypotheses on this agenda is controlled by several factors: how many hypotheses are dependent upon them, the extent of image areas covered by image structures which correspond to the hypotheses, and the global system mode. The ranking of hypotheses on the agenda is itself a rule directed activity. Some of the operations associated with hypothesis instantiation involve task sequences or operations that are performed (e.g., invoking a segmentation procedure). For uniformity, these tasks are also associated with hypotheses in the data base and are ordered on the agenda.

The hypothesis manager determines conflicts in instantiations of hypotheses and evaluates the relative certainty of instantiated hypotheses. This involves monitoring which image structures have not been associated with an object instantiation. This is critical to determining which hypotheses need further elaboration, which should be instantiated, and the global correctness of an interpretation. There will be a graphics-based user interface to the HTDB which enables

displays of the state of the interpretation and interactively determines which operations produced a particular hypothesis and the contexts in which it is valid.

### **2.2.3 Segmentation Knowledge Source**

The Segmentation Knowledge Source consists of rules and strategies that direct the extraction of image structures from the raw images. These are used in two basic modes. One is the bottom-up or data-directed mode wherein the rules extract image structures based upon general perceptual criteria, such as size, regularity of shape and symmetry. The other is a top-down or model directed mode in which the rule application is directed or biased by attempting to instantiate particular types of objects or world relationships. The Segmentation Knowledge Source consists of a library of routines for edge, region, texture, and shape extraction procedures which serve as its basic actions.

Some of the segmentation rules reflect Gestalt goodness-of-form measures in the formation of regions and contours. Simple examples are grouping texture elements together into connected regions and linking edges together under various shape constraints. This knowledge tends to be non-semantic and thus object independent. Other segmentation rules are involved in determining the types of image processing operations to apply to an image given a description of the type of information to be extracted. Other segmentation rules extract and focus processing on significant image structures such as those that are large and homogeneous, or globally connected, or straight, or having constant curvature, or are much different than what is surrounding them. These rules also determine the relations, such as intersection and adjacencies, among such interesting image structures.

The system diagram does not completely describe the relation between the Segmentation Rules and the SAR Object Knowledge Source. The SAR Object Knowledge Source refers to image structures and hypothesis that the Segmentation Rules produce or extract. It also can invoke the Segmentation Rules in attempting to instantiate objects. Thus, the shape and contrast of a country road is specified as queries to the ISDB or could involve invoking a segmentation rule for boundary tracking that is parameterized with contrast and curvature criteria reflecting a country road. In such model-driven processing particular segmentation rules can be applied in restricted image areas to determine predicted

image structures or relations.

#### **2.2.4 SAR Object Knowledge**

Generic SAR object knowledge describes the image properties of image objects such as rivers, roads, bridges, etc. in their various forms, and indicates inter-relationships between the objects. There are many alternative representations of such knowledge based upon such things as object-based descriptions, rule systems, and predicate logic formulations. We have concluded that these choices are generally interchangeable and that the fundamental question is what is represented and how it is organized. For example, the distinguishing characteristics, especially for linear features, are often contextual and involve the relations among instantiated objects. Examples would be the support from an instantiated road-network to enhance/infer the identity of an ambiguous line segment as being a road. This implies the need for representing nested classes of objects to avoid the combinatoric difficulties of describing every possible relationship among all objects. In addition, a representation should support basic class inheritance, similarity, and part-of component relations among objects.

We found it useful to consider a sequence of progressively complicated object representations. A basic one is the use of feature vectors to describe objects. In their simplest form, they are a list of required attributes for some object to be uniquely identified. Feature Vectors correspond to simple queries to the ISDB and the HTDB specific to a particular object or relationship, such as a river segment having particular shape, intensity, and textural properties. Feature vectors can be extended to a frame-like representation wherein the vector components are treated as frame slots for pointers to other frames describing objects and relations. Next would come programmed finders associated with particular objects corresponding to a framed-based representation with procedural attachments. These are a more general object based representation with several procedural attachments, such as explicit strategies for instantiation of an object; predicted adjacency and connectivity properties, and explicit rules for evaluating the certainty of an instantiated hypothesis from distinguishing conditions. Finally, a general reasoning and modeling system would be able to generate and parameterize specific finders from environmental descriptions of objects.

We also distinguish between generic and specific SAR object knowledge. Specific SAR object knowledge represents actual instances of objects in the environment as might be determined from a terrain map. The availability of such information provides many strong constraints on the range of potential interpretations for a given object due to restrictions on its image appearance. These constraints considerably simplify the required representation and inference techniques.

### **2.2.5 System Controller**

The system controller has several high-level executive tasks. One is to interpret user requests into operations that can be performed by the system. It also contains explicit knowledge about global modes of processing such as how to initialize the system for particular types of imagery.

The System Controller acts as an interface between a user and the totally automated version of the system by interpreting tasks into activities of the Segmentation and SAR object knowledge sources. It contains meta-knowledge for different global modes of system processing and monitors the status of an interpretation from the set of instantiated hypotheses and their evaluated certainties. This enables it to determine when the system is stuck and the focus of attention requires alteration or a different mode of processing is required.

Finally, it should be apparent that control is distributed through-out this system. In particular, there is control associated with the instantiation of the generic object knowledge and the segmentation rules. Each monitors the image structure and hypothesis space data bases along with particular foci-of-attention established by the hypothesis manager and or the system controller in determining which rules or objects to instantiate.

## **2.3 PROCESSING SCENARIO**

To better understand the components and implications of our system design, we now consider a simulated scenario, based upon interactive use of ISDB and the image processing techniques we implemented. Initially the system is presented with the image in Figure 2-2 and no a priori information except that this is an aerial SAR image consisting of terrain features and objects for which



**Figure 2-2: Example SAR Image: ETL6**

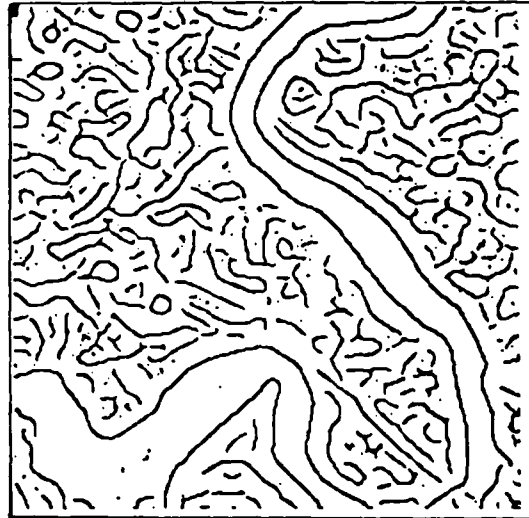
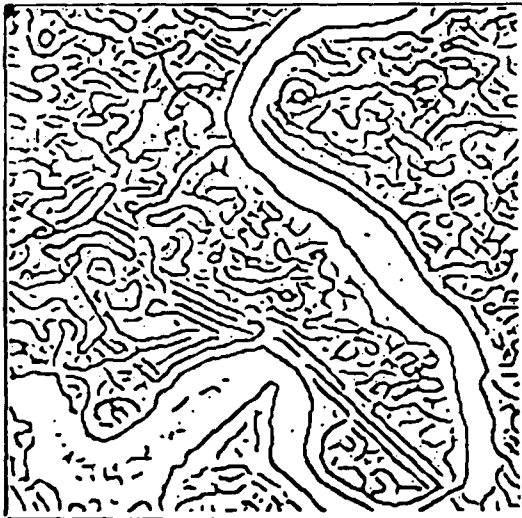
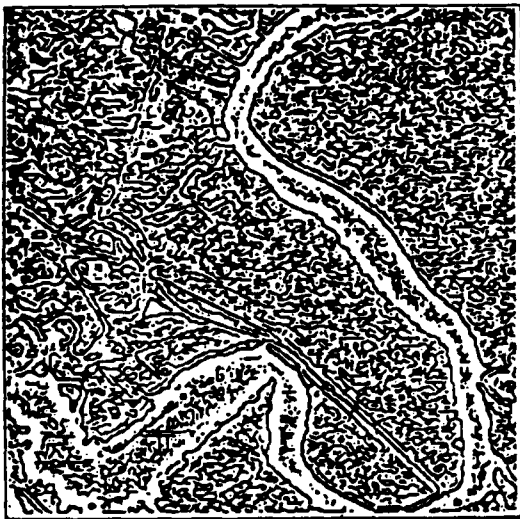
the system has models. In this situation, initial processing is almost totally data-driven until object models can be hypothesized and aid in generation of predictions. This is represented as an explicit mode of processing by the system controller, biasing segmentation processing towards extracting large interesting image structures, which can be accessed by the SAR Object Knowledge Source to instantiate object hypothesis, and generate predictions from expert world object relations and compatibilities. Other processing modes would be for verification of a detailed terrain map and image-to-map registration.

The first stages of processing are directed by segmentation rules that extract and recognize interesting image structures. These extract such things as long connected straight curves at several spatial frequencies, and large regions of homogeneous characteristics. There are explicit criteria of interesting structures, and the system will continue to apply related segmentation rules until a sufficient number of such structures are generated and distributed uniformly across the image. Interesting structures also involve relationships among themselves, such as repetition, symmetry, being parallel, or meeting at right angles. Some of the structures involve global shape characteristics such as curve segments being organized in grids or a radial pattern.

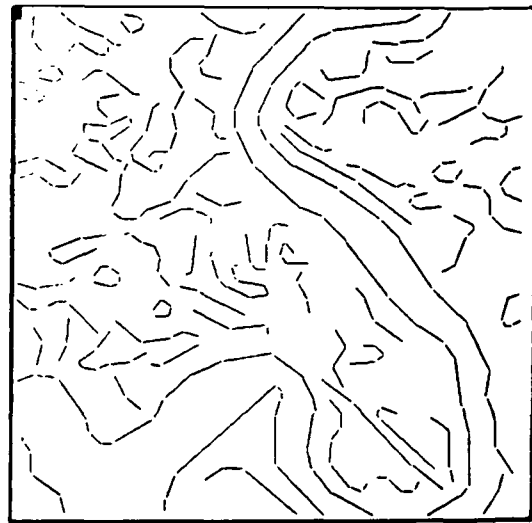
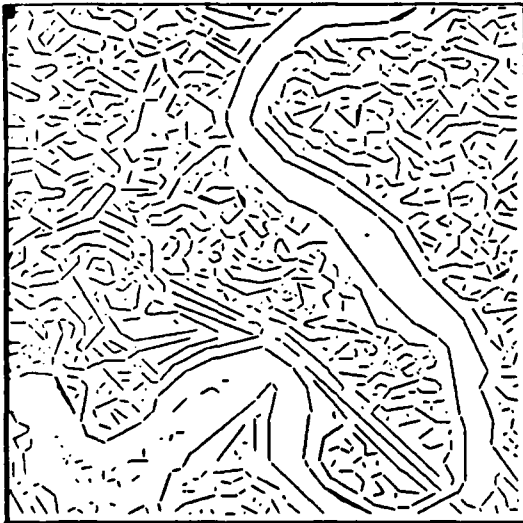
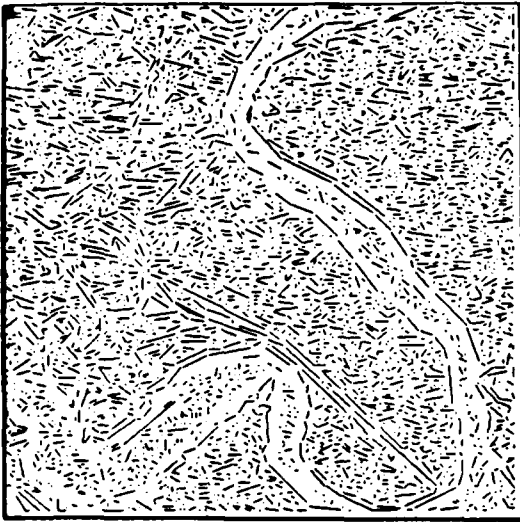
Such segmentation rules generate the initial structures seen in Figures 2-3 through 2-6. Figure 2-3 shows the long connected edges extracted at several different spatial frequencies. Figure 2-4 shows the linear segment approximations to these curves. Figure 2-5 shows the histogram with respect to intensity which was interesting because of its clear bimodality and correspondence to large regions in the image. Figure 2-6 shows the extracted long connected segments.

Each of these extracted interesting structures correspond to entities and relationships in the ISDB. Each such structure is also instantiated as a hypothesis of type INTERESTING-IMAGE-STRUCTURE in the HTDB with associated attributes describing how it was extracted and by what criteria it is interesting. The importance of the extracted structures on the Agenda is determined by attributes such as size, and potential attachments to SAR object formats.

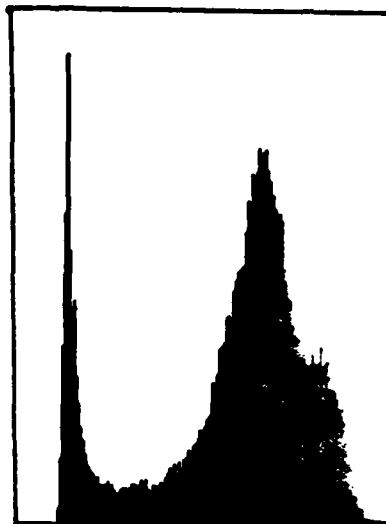
When a sufficient number of interesting image structure hypothesis are generated, the SAR object knowledge source begins generating object hypothesis by matching attributes of the interesting structures and those associated with the



**Figure 2-3: Contours at Different Spatial Frequencies**



**Figure 2-4: Linear Segment Approximations**



**Figure 2-5: Intensity Histogram and Extracted Image Areas**



**Figure 2-6: Long Connected Contours**

object models. Initially it is biased toward matching to the structures that are largest, the most regular, and for which the object attribute matches are strongest. In this case, the largest structures that can be reliably matched are the LAND-TERRAIN-AREA and LARGE-RIVER-AREA. The LARGE-RIVER is indicated by attributes of being dark, large, and elongated. LAND-TERRAIN is indicated by multiple contrasts at high density.

The SAR-object-network, part of the SAR Object Knowledge Source in Figure 2-1, stores the general object types that are compatible with a LARGE-RIVER. Associated with the models of these associated objects are explicit finders that will direct queries to the ISDB and focus application of segmentation processes to the image. This is constrained by the instantiation of the LARGE-RIVER hypothesis. These begin looking for riverbanks (elongated bright or dark regions parallel to the boundary of the river), bridges (roads or long straight lines roughly perpendicular to the river), and tributaries (windy, dark regions leading off of the river). The certainty of the LARGE-RIVER object is associated with the success of the finders associated with its compatible or component objects.

Some of the actions associated with the Finder for tributaries from large rivers are shown in Figures 2-7 through 2-12. Figure 2-7 shows a close-up view of the high frequency curve segments extracted from the image in Figure 2-2. The Finder looks for a high contrast edge near the boundary of the extracted river segment shown in Figure 2-8. A selected edge segment is shown in Figure 2-9 along with the attributes associated with that edge in the image structure data base. It then evaluates the average intensity across this segment and generates a binary image at this average intensity. The resulting image is evaluated for long, connected, winding regions which are connected to the LARGE-RIVER-AREA (Figure 2-10). If none are found, a different high contrast segment is selected. The boundaries of the binary image (Figure 2-11 and 2-12) are then used to direct an edge-linking process to follow parallel curve segments near the boundaries of the binary image.

The Bridge-Finder looks for long straight regions or curves that are not aligned with the river and intersect anomalies, such as bright spots in the LARGE-RIVER region. It can also extend the curves across the river looking for a continuation of the curve segments on the other side. Figure 2-13 shows the extracted edge segments from the lower right hand corner of the image. Figure

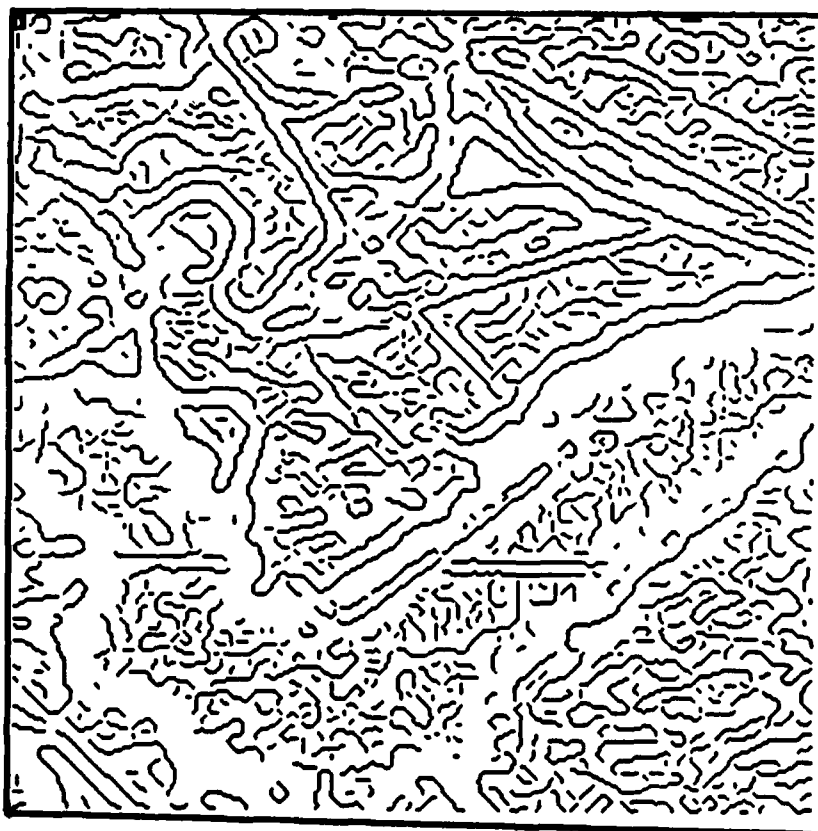


Figure 2-7: Image and Extracted Edges

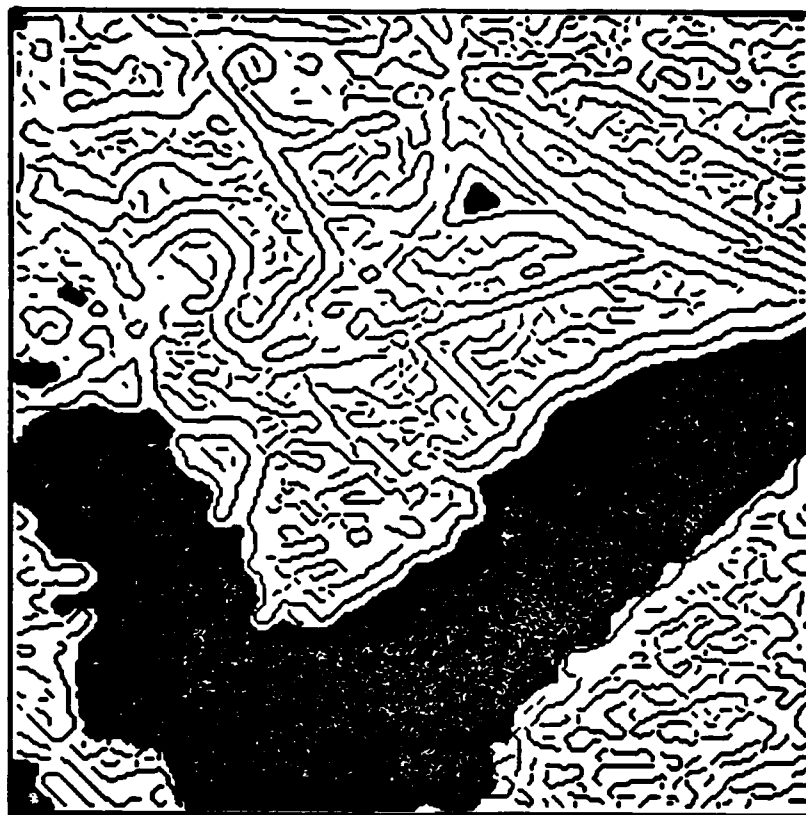


Figure 2-8: Histogram Extracted River Superimposed

```

#<EDGE 13272724>, an object of flavor EDGE
'
has instance variable values:
FROM-IMAGE:      20
LABEL:           2040
LENGTH:         6
PT1:             (54 121)
PT2:             (49 117)
COORDINATES:     ((54 121) (53 1
20) (52 120) (51 119) (50 118) (49 117))
DECOMPOSITION-LIST:  NIL
ORIENTATION:     (-0.8 1 77.8)
LINEAR-DEVIATION: (0.2195122 2 (5
2 120))
AVERAGE-CONTRAST: 41.625942
AVERAGE-INTENSITY: 127.46053
#<EDGE 13272724>
(clear)
NIL
(display-edge 2040)
NIL

```

Figure 2-9: Selected Edge and Attributes



**Figure 2-10: Selected Edge and Corresponding Binary Image**



**Figure 2-11: Extended Potential River Regions**



**Figure 2-12: Boundary Constraints on River Tracking Procedure**

2-14 shows the river area overlaid with the thinned version of these edges. Figure 2-15 shows the long straight edges, which are not aligned with the river and also intersect the river at an anomolous bright spot corresponding to a bridge.

An anomaly is any interesting image structure that is not associated with an instantiated object hypothesis or an object that is incompatible, as specified in a network of relations between world objects, with an instantiated object. Finders can also exist for particular types of anomalies, such as unaccounted objects in the river. Figure 2-16 shows the extracted high frequency contours in the river area. Figure 2-17 shows which of these contours exceeds a minimal length criteria and Figure 2-18 shows the straightest subsegments associated with these. These (Figure 2-19) are anomolous structures in the river. Such structures are compatible with bridges or boats but the finders for these objects would be unsuccessful since the associated contours are too dark and large to be boats and are not bridges, since they are not oriented with any significant linear structures on the land areas. These are high frequency, very low contrast features which would be removed by the noise estimating process associated with the edge operator used in their extraction. By basing their extraction on structural criteria however, we are able to find them. Anomolies will often correspond to world objects for which the system has no a priori model.

In parallel with the Finders and other potential, compatible objects activated by the LARGE-RIVER-AREA hypothesis, are Finders and objects associated with the TERRAIN-LAND-AREA. In fact, the control of hypothesis verification and generation becomes more and more decoupled as distinct image areas are partitioned. This is reflected by the System Controller allocating different resources to different parts of the image during processing if there are multiple processors. Terrain types are distinguished by textural classification into such types as URBAN, FOREST, SUBURBAN and further subtypes. There are also perceptual textural typing associated with particular segmentation rules. Figure 2-20 shows the extracted high frequency contours in the image. Figure 2-21 shows the selected short linear segments extracted from these image contours and restricted to the TERRAIN-LAND-AREA. Such edges tend to form a useful set for computing textural properties with respect to their average contrast, orientation, or alignment with respect to a local neighborhood. Associated with each TERRAIN-LAND-AREA subtype are feature descriptions parameterized by sensor parameters. The histogram with respect to the average contrast computed

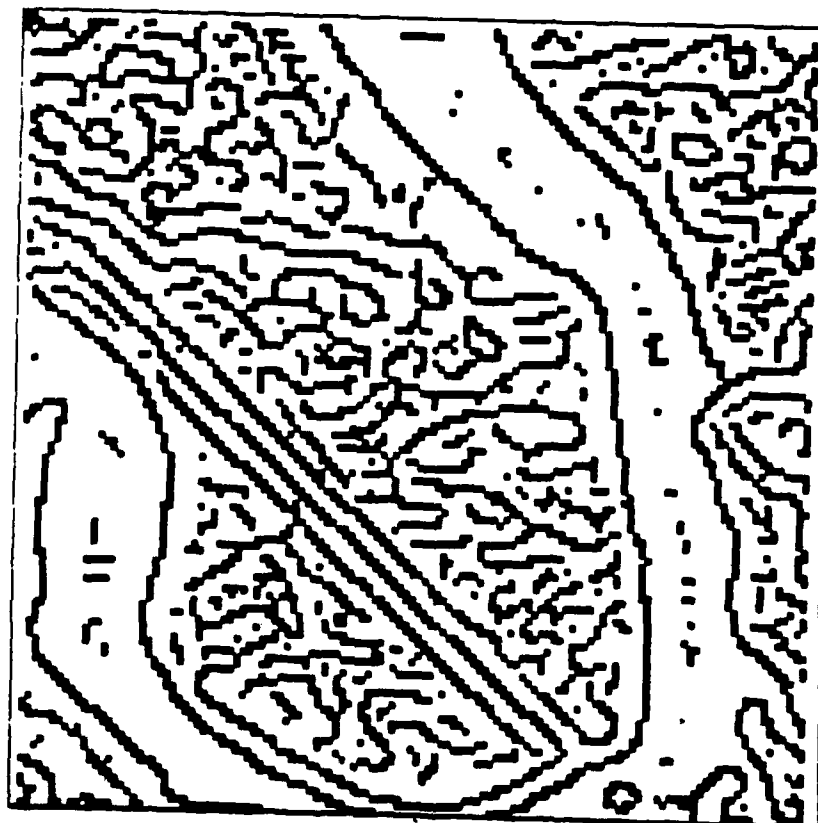
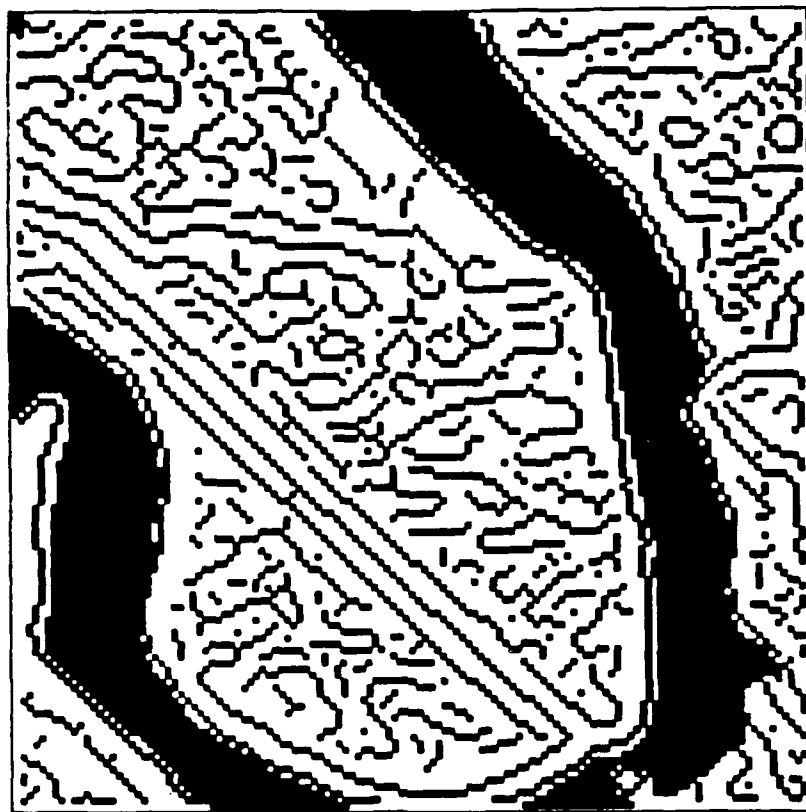


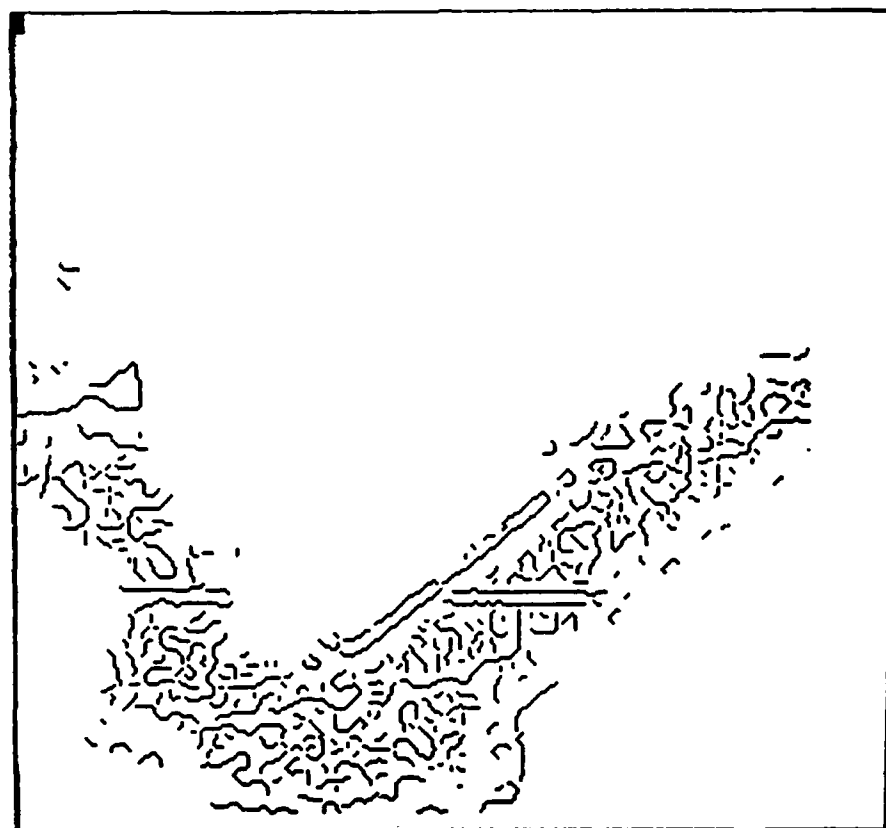
Figure 2-13: Image and Extracted Edges



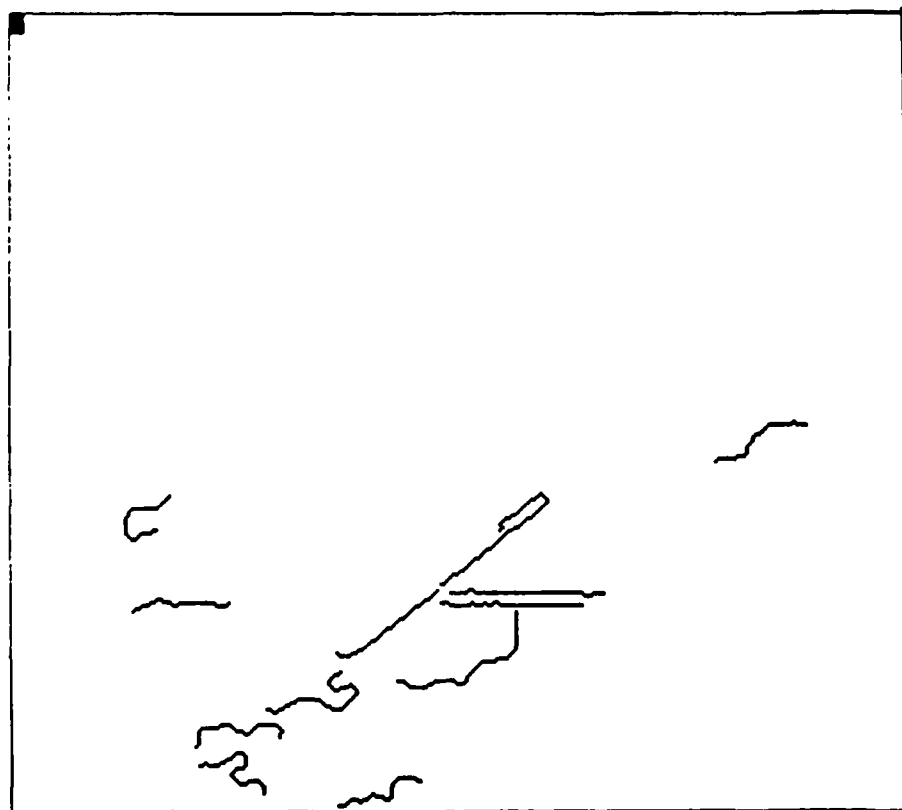
**Figure 2-14: Overlaid River**



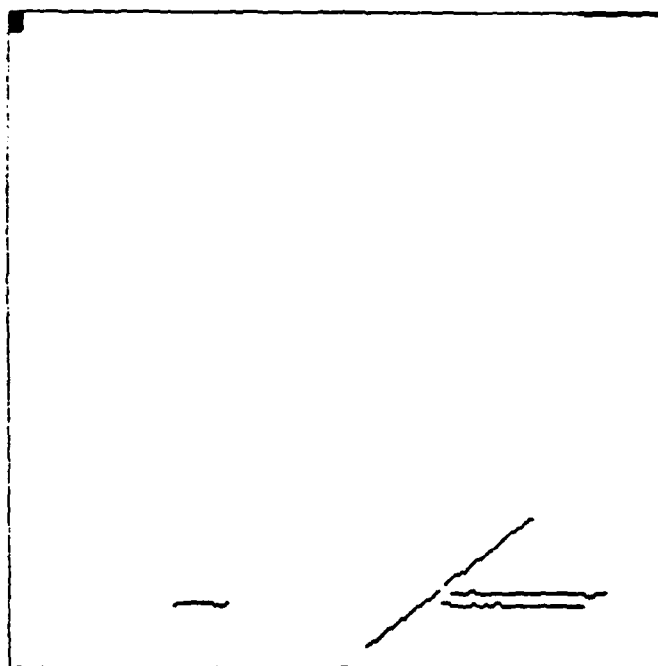
**Figure 2-15: Long Linear Segments**



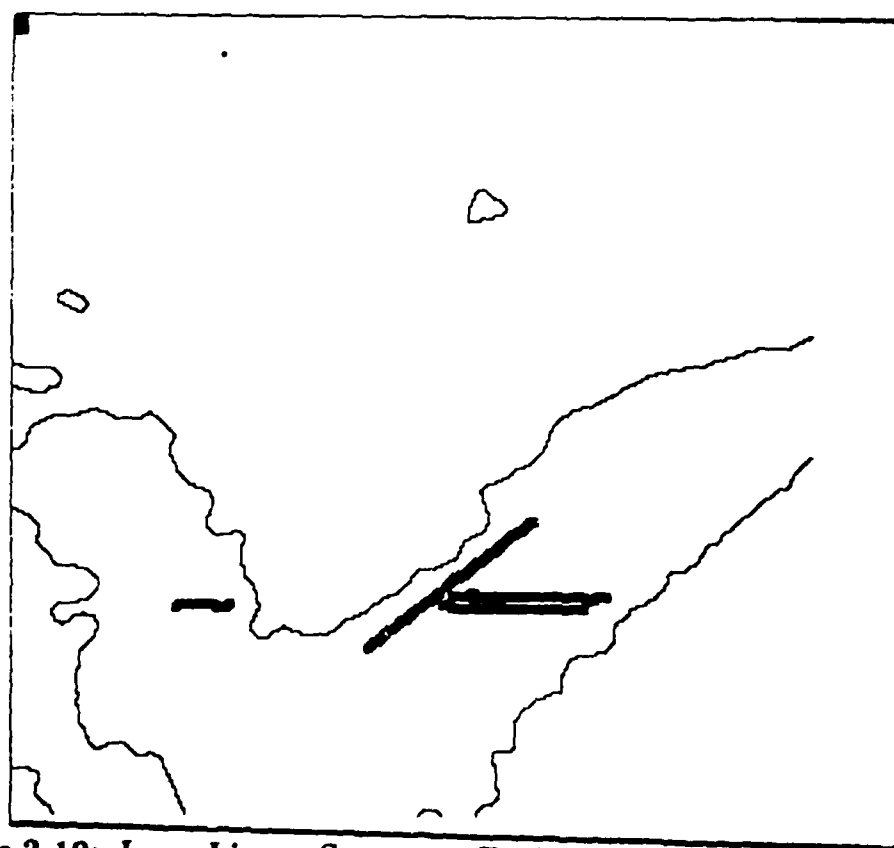
**Figure 2-16: Edges in River Area**



**Figure 2-17: Long Edges in River**



**Figure 2-18: Long Linear Segments**



**Figure 2-19: Long Linear Segments Positioned in River Region**

along the selected segments shows a distinctive peak (Figure 2-22) which maps onto the curves shown in Figure 2-23. The remaining segments are shown in Figure 2-24. The thresholded density plot of the segments in Figure 2-24 are shown in Figure 2-25. The properties of dense, randomly oriented texture, at low intensity correspond to potentially forested areas. FORESTED-AREAS in turn, activate Finders for tree-lines, indicated by long contours that are very bright on one side and dark on the other side.

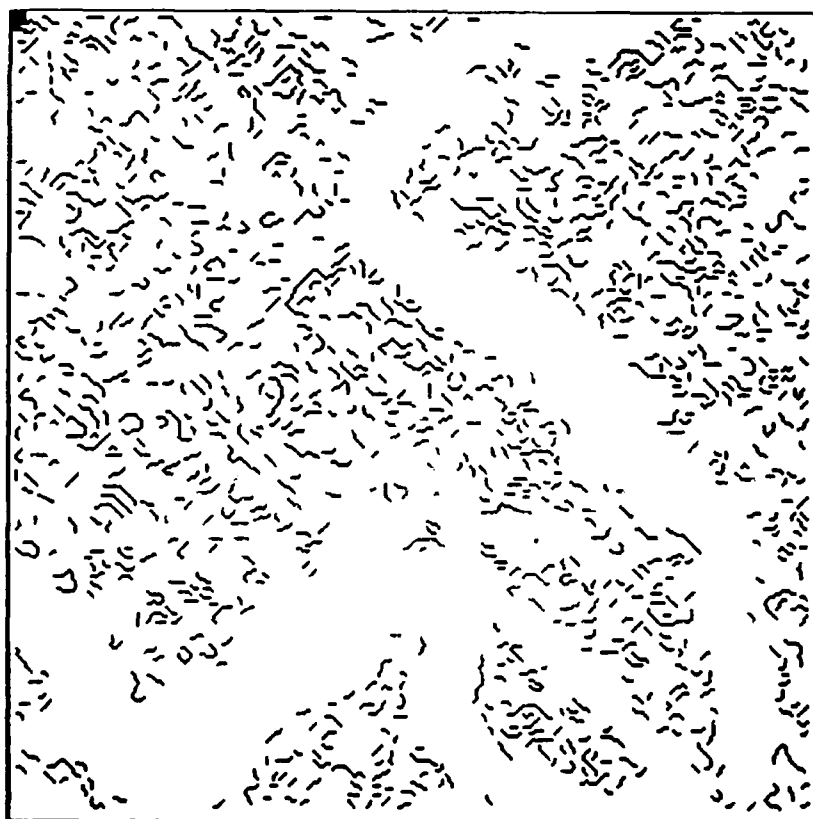
The non-forested image areas are evaluated with respect to other generic terrain types such as urban or agricultural. Urban terrain is indicated by high contrast, orthogonal texture elements. Figure 2-26 shows an enlargement of the upper left hand corner of the original image. Figure 2-27 shows the orientation histogram of this with respect to the linear segment approximations thresholded with respect to contrast. A non-uniformity of texture element orientation is indicated by the distinct peaks. Figure 2-28 shows the image segments correspond to the large peak on the right. Figure 2-29 shows a histogram with respect to those selected elements. The texture elements corresponding to the two large, roughly orthogonal peaks in this histogram are shown in Figures 2-30 and 2-31.

In the non-forest, urban areas finders for instances of attributes for buildings, roads and patterns are applied. The road in this area of the image is somewhat interesting because large segments along it are obscured. The Road-Finder executes a set of segmentation procedures biased to find long connected segments perpendicular to the grid orientation. Figure 2-32 shows the extracted edges and Figure 2-33 shows the linear segment approximations to those edges which exceed some threshold with respect to length. Figure 2-34 and 2-35 show a set of linear segments selected by an edge-tracking procedure which was initialized with the uppermost edge in the set. This tracking was based upon maintaining a smoothly changing orientation with similarly oriented contrast for each selected edge. This corresponds to the attributes of a road parameterizing an edge tracking procedure. For rivers, the tracking procedure could be initialized to track both boundaries simultaneously with global windiness allowed in the orientation changes. Figures 2-36 and 2-37 show the connected contour with respect to the linear segments and edges.

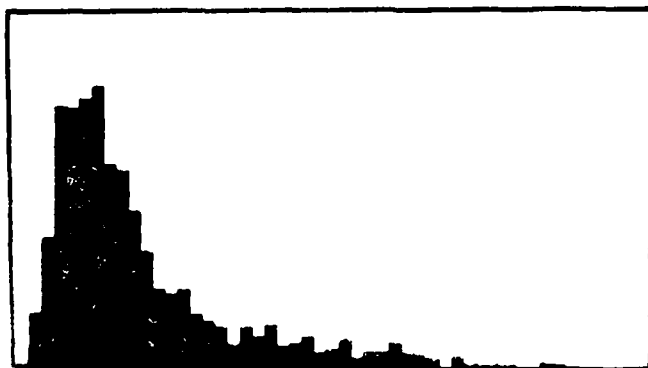
At this stage of processing, the system has determined the basic terrain types and parts of the river network. It has also determined basic objects and



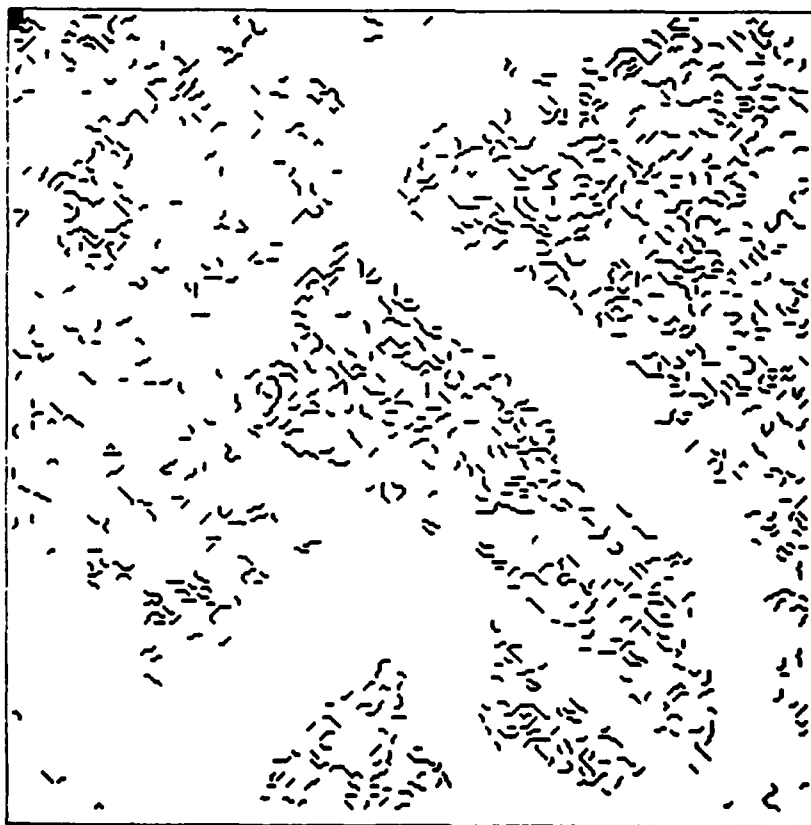
**Figure 2-20: Extracted Edges**



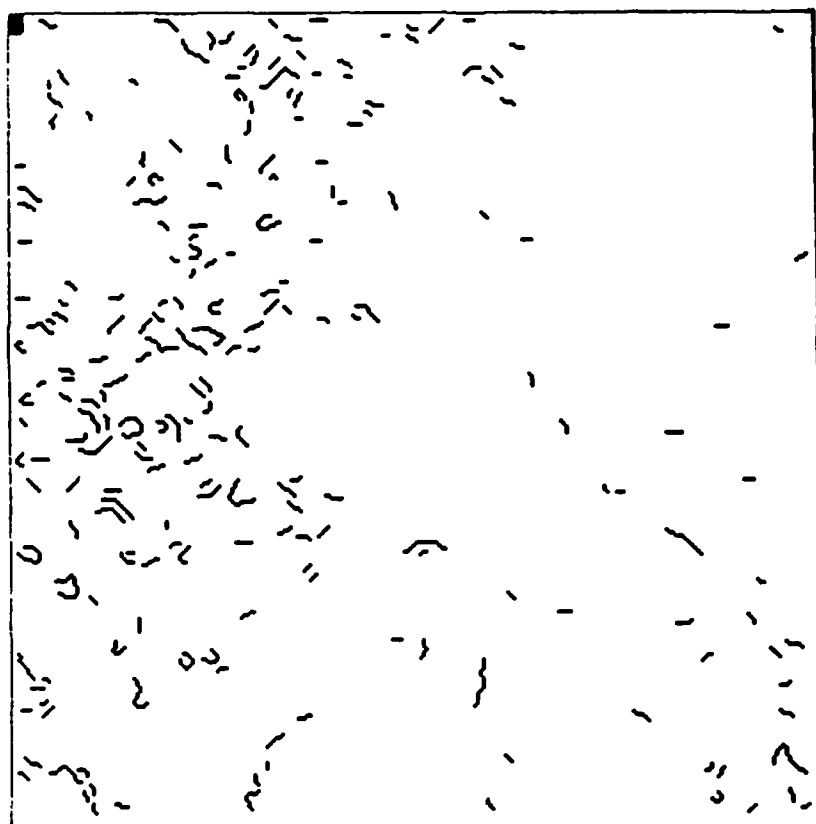
**Figure 2-21: Short Linear Subsegments**



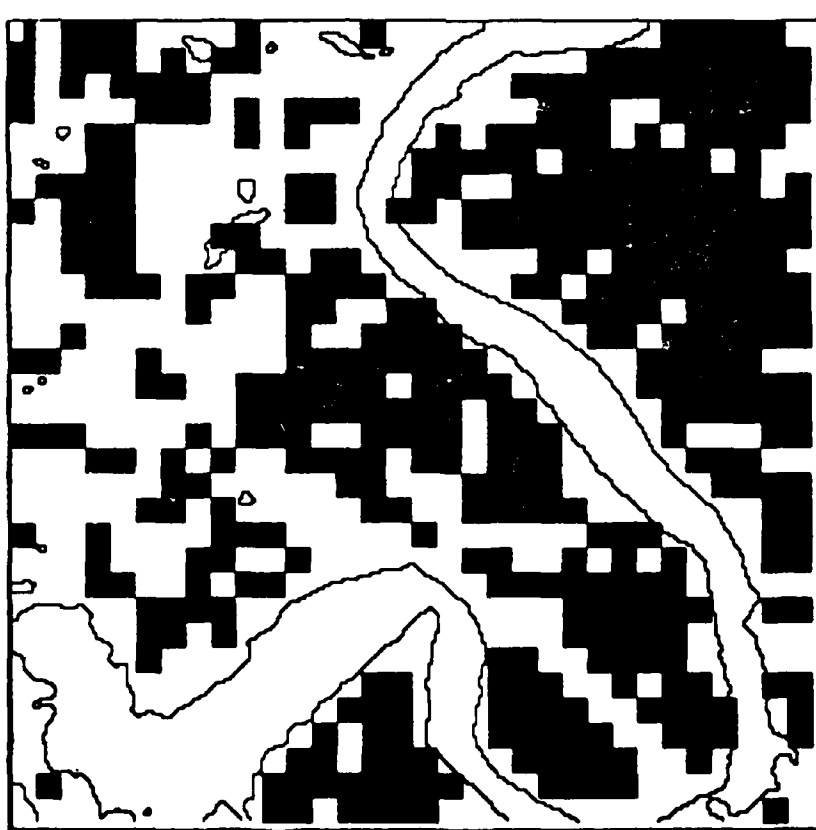
**Figure 2-22: Average Intensity Histogram for Edge Segments**



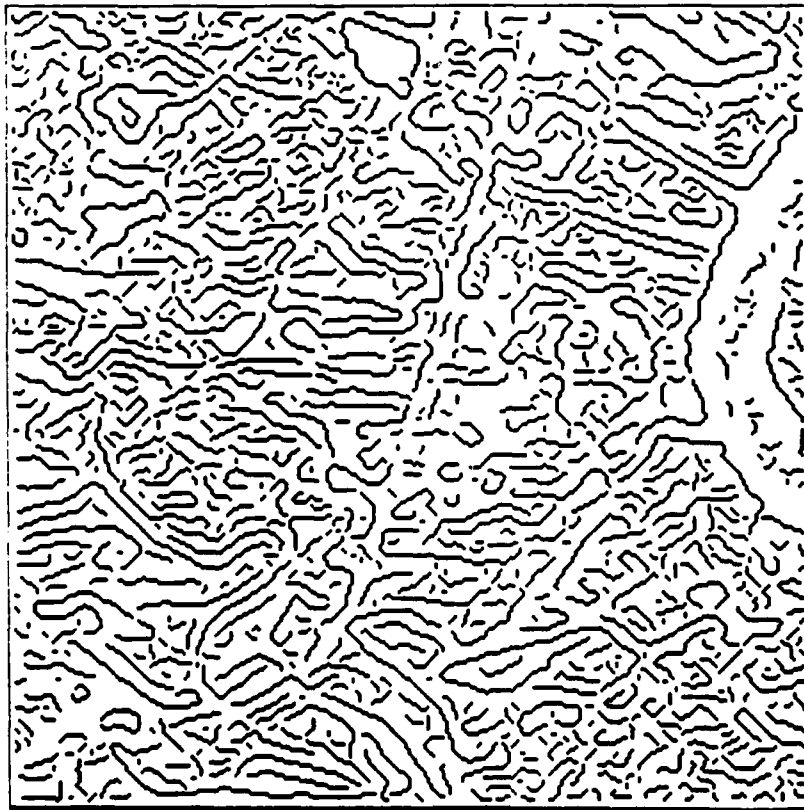
**Figure 2-23: Selected Edges Corresponding to Major Cluster**



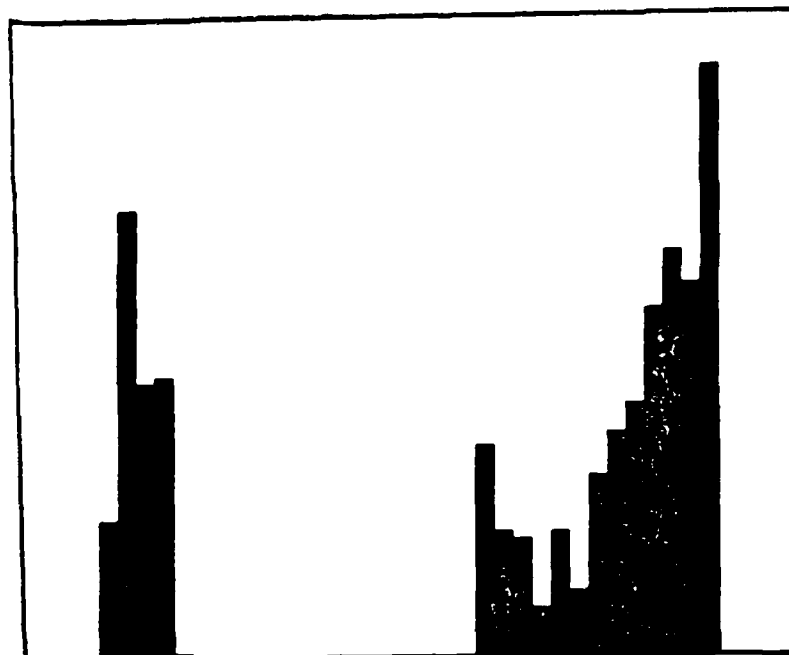
**Figure 2-24: Other Edges**



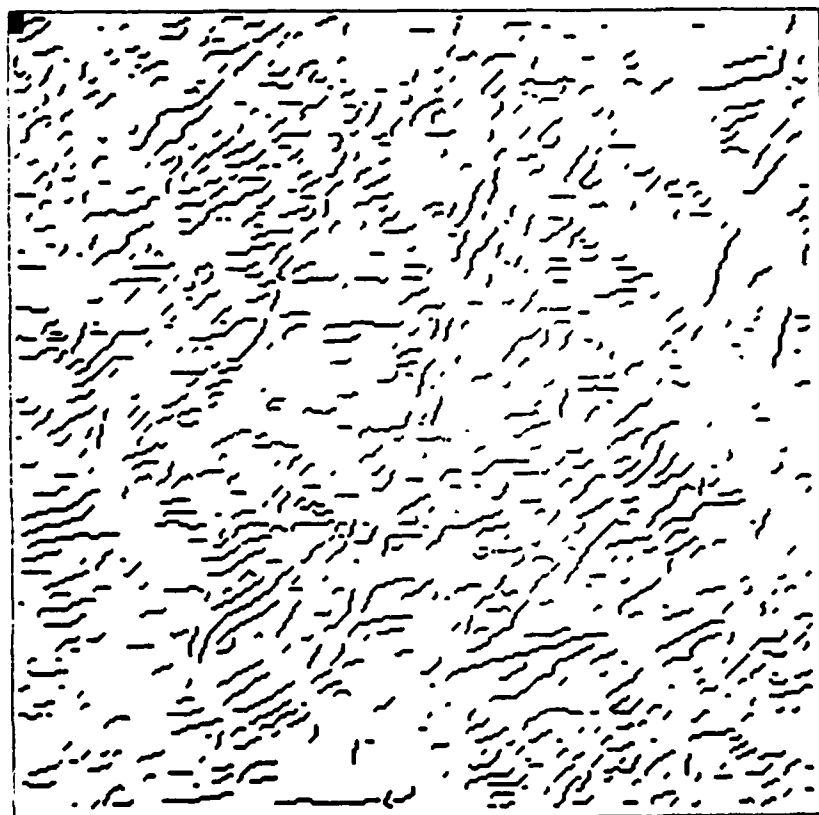
**Figure 2-25: Thresholded Density Plot of Edges from Peak**



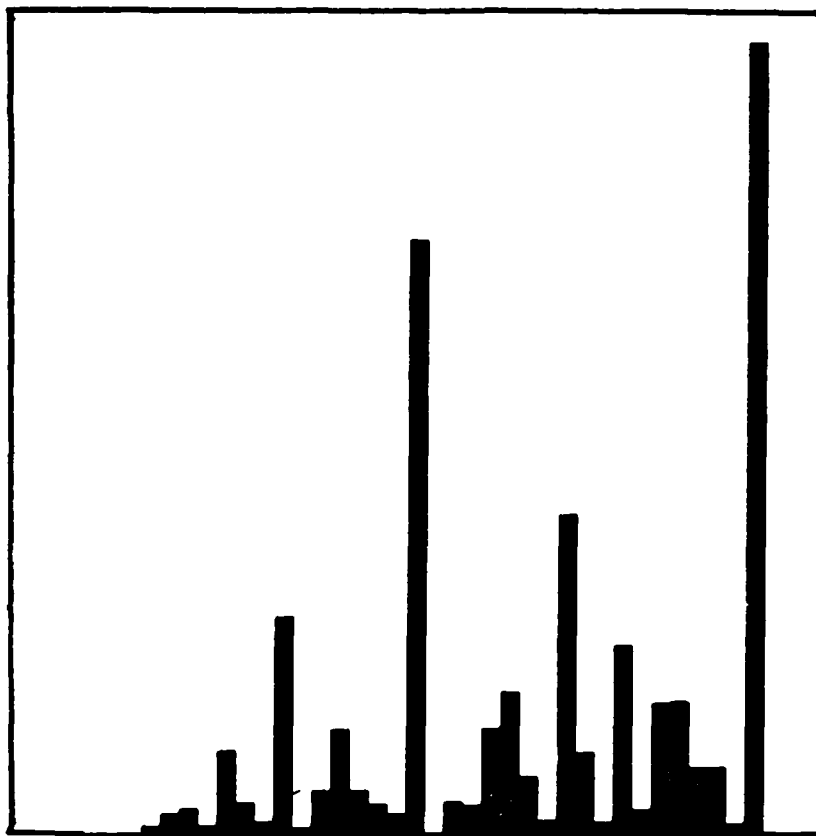
**Figure 2-26: Image and Extracted Edges**



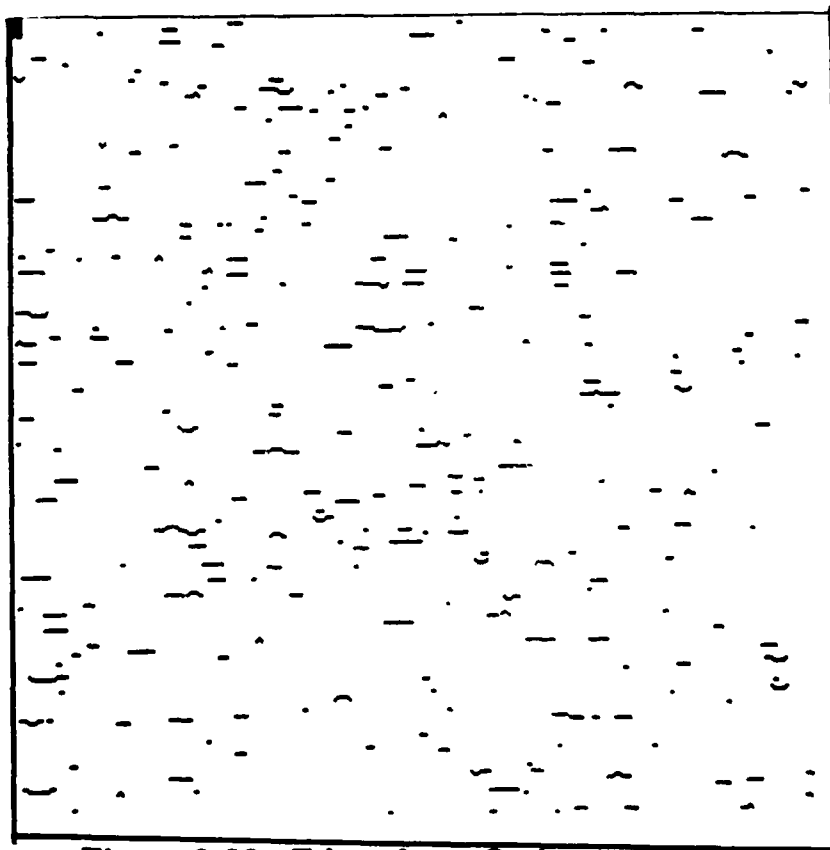
**Figure 2-27: Orientation Histogram**



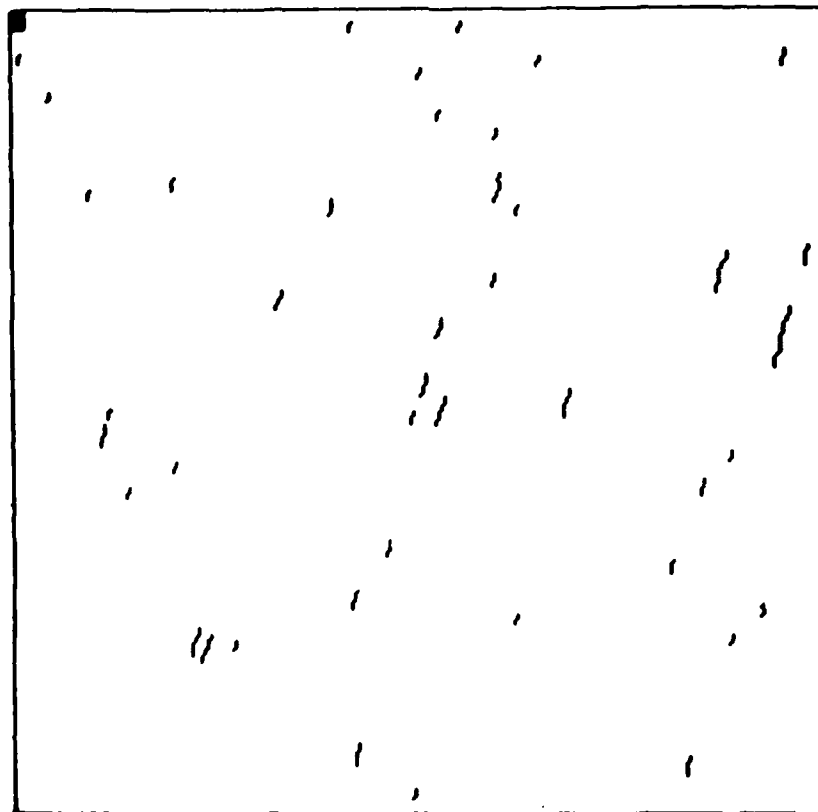
**Figure 2-28: Edges Mapped from Major Cluster**



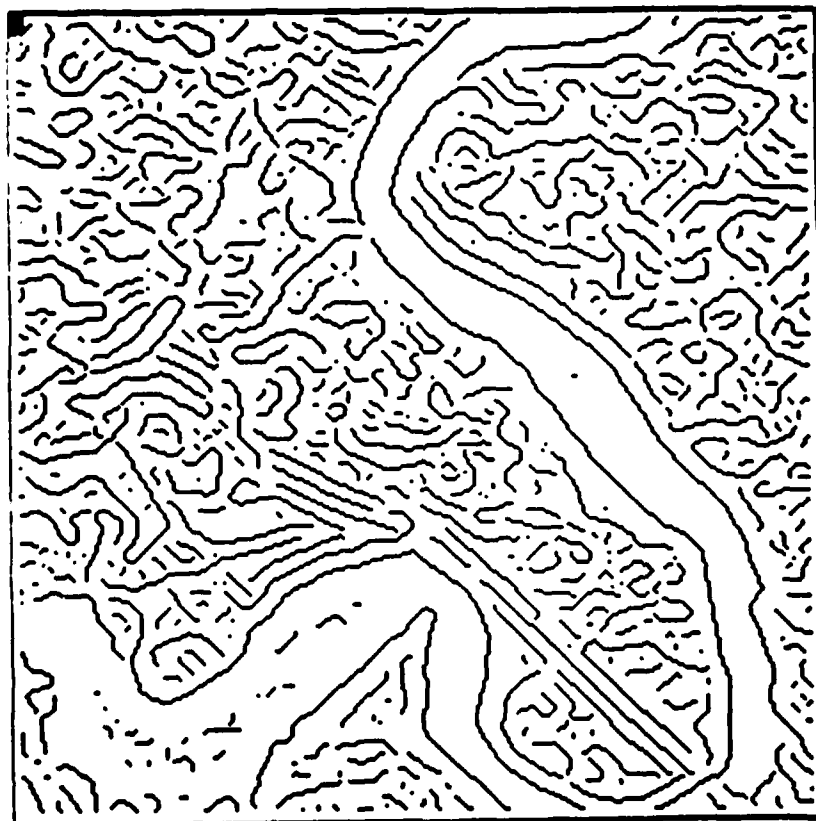
**Figure 2-29: Orientation Histogram from Selected Edges**



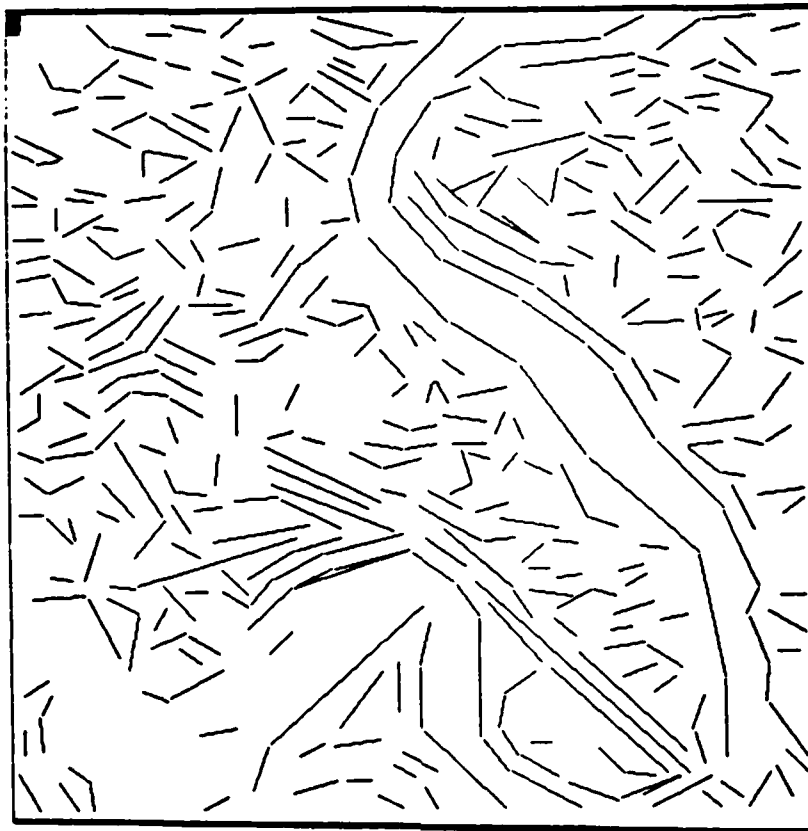
**Figure 2-30: Edges from Orthogonal Peaks**



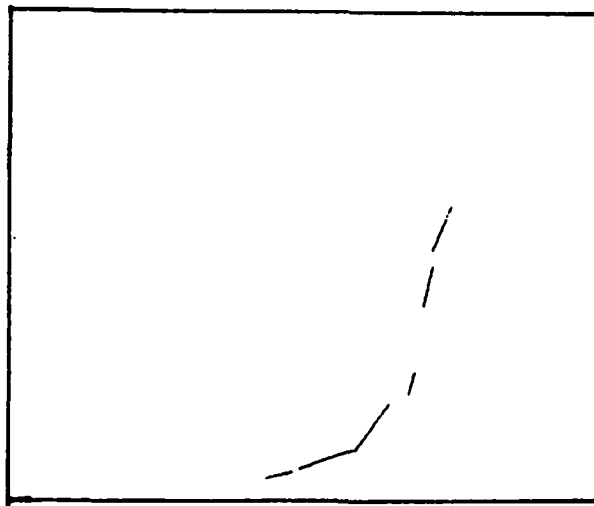
**Figure 2-31: Edges from Orthogonal Peaks (cont'd)**



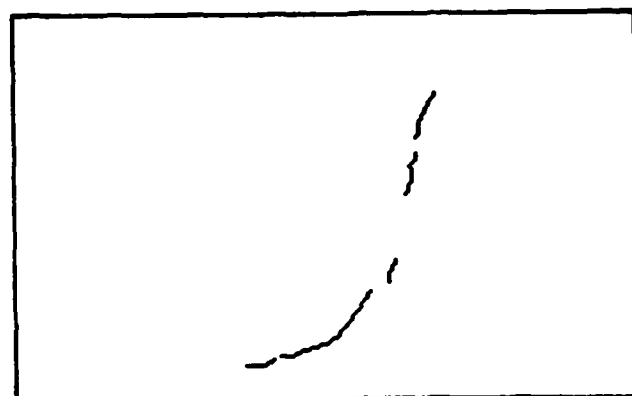
**Figure 2-32: Extracted Edges**



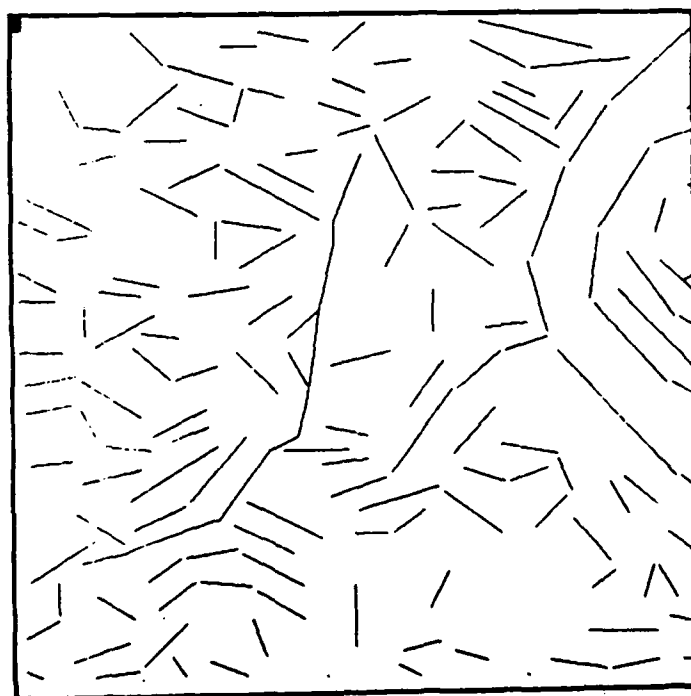
**Figure 2-33: Threshold Linear Approximations**



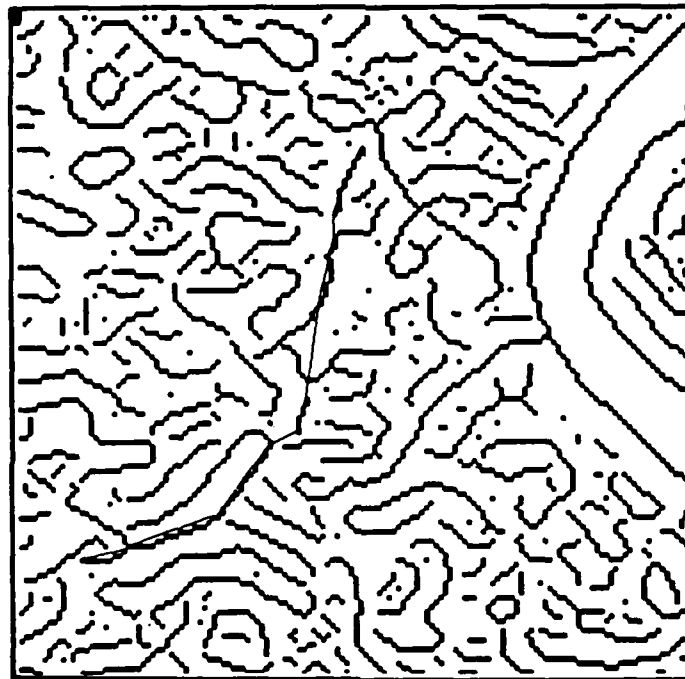
**Figure 2-34: Linked Edge Fragments**



**Figure 2-35: Linked Edge Fragments (cont'd)**



**Figure 2-36: Linked Edge Fragments  
Superimposed on Extracted Edges**



**Figure 2-37: Linked Edge Fragments  
Superimposed on Extracted Edges (cont'd)**

features associated with these: roads, tree-lines, and bridges. Basic to this processing is the network in the SAR Object Knowledge Source which describes general object relationships. It directs processing through the use of contextual information: the bright blob in the river area is processed differently than those in terrain areas. Processing can continue to finer and finer levels of detail from the predictions generated from each instantiated object's expected relationships to other objects. Processing also continues by matching object format descriptions against extracted image structures which have not been reliably associated with an object hypothesis or by attempting to resolve conflicting object hypothesis which are associated with the same or intersecting object hypothesis. In this case, the predicted relations associated with the object type descriptions for the conflicting hypothesis are used to direct the disambiguation.

### 3. IMAGE STRUCTURE DATA BASE

The Image Structure Data Base (ISDB) is where all the basic operations for representations are applied to and processing results obtained from a set of images and related structures. In this section, we describe the objects that are represented in the ISDB, their attributes, the types of queries and operations that may be applied over these objects, and aspects of the implementation of the ISDB in ZETA-LISP FLAVORS. The basic role of a symbolic/relational data base describing extracted image information is well established in computer vision systems. There are many spatially tagged, symbolic representations used in image understanding systems: the primal sketch of Marr [Marr - 82], the curvature primal sketch of Asada and Brady [Asada - 84], the RSV structure of the VISIONS system [Hanson - 78a,b] the patchery data structure of Ohta [Ohta - 80], and Haralick's [Laffey - 82] topographic classification of digital image intensity surfaces. These all map the results of various image processing routines into a symbolic, image-registered data base that is accessed by the different types of system knowledge. Generally, the recognition of objects and more complicated image structures are expressed as grouping operations over queried entities in these image structure data bases. Our implementation of the ISDB also has the capability of representing non-image processing results, processing history, and binding these to instantiated rules and object formats.

#### 3.1 OBJECTS

The ISDB is implemented as an extendible set of objects common to *object-oriented programming* [Goldberg - 84, Krasner - 83 and Moon - 84]. In this, we explicitly define an object type, its attributes, and the operations that can be performed upon it. This style of programming supports modularity and inheritance of attributes and operations over different, but related, object types. In the ISDB, there are object types for images and basic image structures such as regions, curves, and points. Relationships between structures, such as ADJACENT-TO or CONTAINS, are also represented as objects as are non-image structures, such as the descriptions of processing steps, tables, relational networks, and histograms. Particular instances of an object type are said to be *instantiations* of the general object type.

Our implementation corresponds directly to the FLAVOR mechanism used in SYMBOLICS ZETA-LISP. Our objects are implemented as flavors with queries implemented as methods and functions over these flavors and inheritance of attributes and methods by FLAVOR-mixing.

### 3.1.1 Images

A basic object is an image. An image has the following attributes:

#### IMAGE

- NAME:
- DOCUMENTATION:
- HISTORY:
- FORMAT:
- ARRAY-TYPE:
- SRC-FILE:
- DIMENSIONS:
- IMAGE-STATISTICS:
- ARRAY

NAME: is the how the image is referred to in the active system. This can either be a unique string or number. DOCUMENTATION: specifies a text file describing any aspects of the image that a user cares to. HISTORY: is the sequence of operations that were performed in the production of the image. This list is updated automatically whenever a function is applied to an image. SRC-FILE: specifies where the image is secondary storage. If this is nil, then the image has not been saved. ARRAY-TYPE: specifies the type of the array storing the pixel values. DIMENSIONS: is a list of the x and y dimensions of the image and any indexing offset that may be used. IMAGE-STATISTICS is a property list containing such things as minimum and maximum value in the image and the variance. Since it is a general property list, it can be extended with additional attributes. ARRAY points to the array containing the image.

We have also developed a more general construct called a STACK. This is a set of images, which may or may not be of the same resolution. For example, a set of images which are related, such as different bands of some sensor form a stack without resolution reduction between the levels. For pyramid structures such as quad-trees, there is a resolution reduction. The attributes of a STACK are:

#### STACK

- NAME:
- TYPE:
- DOCUMENTATION:
- SRC-FILE:
- NUMBER-OF-LEVELS:
- NEIGHBORHOOD-MAPPING-DOWNWARDS:
- NEIGHBORHOOD-MAPPING-UPWARDS:
- IMAGE-LIST:

The NEIGHBORHOOD-MAPPINGS: specify which pixels are descendents and parents of a given pixel in the  $n$ -th image in the  $n+1$  and  $n-1$  levels of the stack. This is for specifying relative access functions across levels. The IMAGE-LIST: is a list of pointers to the images comprising the different levels of the stack.

#### 3.1.2 Image Structures

We currently represent three different types of image structures: points, curves, and regions. A point is a discrete image position, a curve is a connected sequence of points, and a region is a connected area of points. For each of these, we distinguish between its locational (-LOCATIONAL) properties based primarily upon the positions of the points that particular instances of these objects consist of and to their attributes (-ATTRIBUTES) based upon the image values at these points. These aspects of image structure objects are represented by different flavors in ZETA-LISP [Moon - 84] which are integrated by flavor mixing. Thus, a CURVE inherits the CURVE-LOCATIONAL and CURVE-ATTRIBUTE object descriptions.

The geometric properties of a curve are represented by the CURVE-LOCATIONAL Object definition. This is defined as:

#### CURVE-LOCATIONAL

- STRAIGHT:
- OPEN:
- LENGTH:
- PNT1:
- PNT2:
- POINTS:
- SHAPE:
- GRID:

Many of these attributes are related and described by a single number while others are structured property lists which themselves consist of instances of other objects. STRAIGHT: is a logical variable describing whether the curve is completely described by the positions of its endpoints. OPEN: is a logical value describing whether the curve is a loop or not. LENGTH: corresponds to the number of pixel-steps along the curve. PNT1 and PNT2 are the endpoints of the curve. These are not set if the curve is not open. POINTS is the list of the image-position coordinates sequentially ordered. Note that the value of LENGTH is the number of elements in the list associated with the attribute POINTS. SHAPE is a property list consisting of the different types of shape descriptions that are used in describing the shape of curves. These shape descriptions will in general be objects defined by flavors in ZETA-LISP. Some of the curve shape descriptions used are the sequence of curvature approximation values along the curve; contour orientation histograms, and decomposition into linear sub-segments. Note that the same shape description may have different properties depending upon the parameters used in the shape extraction processing. An example of the property list associated with the SHAPE attribute would be:

## SHAPE:

```
((Linear-Approximations
  (#<LINEAR-SEGMENTS 25623674>
   #<LINEAR-SEGMENTS 25626617>
   #<LINEAR SEGMENTS 25630327>))
 (Contour-Histograms
  (#<CONTOUR-HISTOGRAM 25612226>
   #<CONTOUR-HISTOGRAM 25617217>)))
```

The value of the property Linear-Approximations is a list of instances of the object type LINEAR-SEGMENTS, which correspond to different piecewise linear decompositions of a curve:

```
LINEAR-SEGMENTS
  DECOMPOSED-CURVE:
  CURVE-LIST:
```

where DECOMPOSED-CURVE: points to the curve being approximated and CURVE-LIST is a list of pointers to the instantiated CURVES corresponding to the linear segments. It is possible for the SHAPE: property list not to point to defined objects. Nonetheless, we feel that any image operation should have an explicit type of object associated with it for modularity and system extendibility. The motivation here is to allow for diverse shape descriptions to be associated with curves without adding an endless set of attributes to the object definition.

The parameters describing the extraction of the linear segment approximation are contained in a more general object type, the ISDB-OBJECT type described in Section 3.2. Whenever, an object is generated and placed in the ISDB, how it was extracted is associated with this general object type. Other attributes associated with the ISDB-OBJECT description are ASSOCIATED-RELATIONSHIPS and ASSOCIATED-HYPOTHESES. These are lists containing pointers to all the instantiated relationships that an object is involved with and all the Hypothesis in the Hypothesis Data Base that an object is involved

with, respectively.

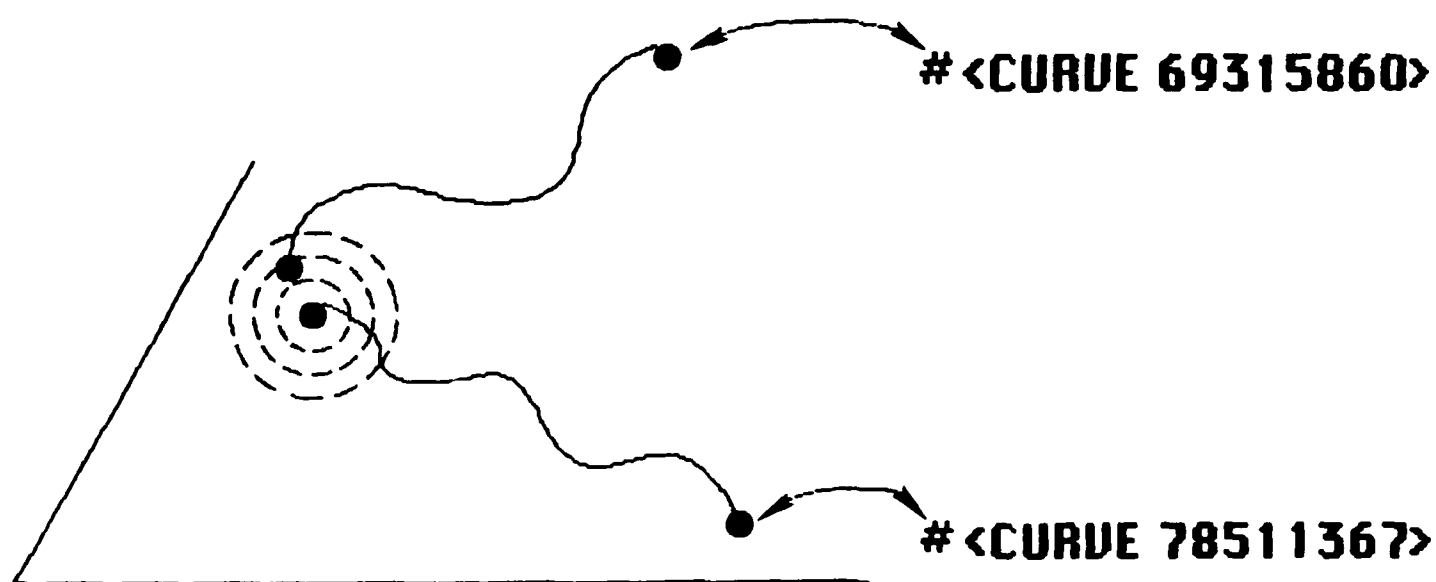
GRID points to the particular image that contains the curve's points labeled as a single connected entity. This results from a connected-components or edge walking process. The label associated with each point along the curve is a pointer the instance of the object describing the curve itself. This is essential: it allows us to do geometric processing on the grid and still be able to associate the results with the instantiated curve itself. For example, if processing requires determining all curves in some area, the pointer-values in the corresponding grid image can be sampled and then stored in a list of curves to which further processing is restricted (Figure 3-1). The same processing can be done to sample the pointer values in other registered images, as to determine all the regions which are nearby a particular curve.

Besides the geometric specification of curves are the attributes determined from the image values at the points they contain. These correspond to properties determined from different images at locations along the curve. Typical examples of these are the average intensity or contrast along a curve and the variance of these things. These are represented as the CURVE-ATTRIBUTES:

#### **CURVE-ATTRIBUTES**

- **CONTRAST-IMAGE:**
- **CONTRAST-AVERAGE:**
- **CONTRAST-VARLANCE:**
- **INTENSITY-IMAGE:**
- **INTENSITY-AVERAGE:**
- **INTENSITY-VARLANCE:**
- **GENERAL-CURVE-ATTRIBUTE-LIST:**

These are mostly self-explanatory. Contrast is the magnitude of the image gradient. The CONTRAST-IMAGE describes the image the contrast values are found in, CONTRAST-AVERAGE describes the average contrast value computed from the points along the curve, and CONTRAST-VARLANCE is the variance of these values. The attributes are similar for image intensity values. The GENERAL-CURVE-ATTRIBUTE-LIST is a property list which allows for



Each point along the curve **#<CURVE 69315860>** is labeled with a pointer to the instance of the curve object **#<CURVE 69315860>**. Determining that **#<CURVE 69315860>** is near an endpoint of **#<CURVE 78511367>** requires sampling the label values in neighborhoods centered on the endpoint. Then the global properties of the curves can be accessed through their object descriptions and further processing based upon this.

**Figure 3-1: Object-Label Grid**

extending the types of attributes that can be computed along a curve. These consist of a list of lists for each such attribute. The sublist contains a pointer to the image the attribute was computed over, and different characterizations of the attribute values along the curve. Examples of other useful curve attributes are the average chamfer values along a curve with respect to some image object and the variance of this, to determine relative position and orientation of image objects. Another is the number of intersections-points of a curve with an extracted region.

Regions are connected areas of images. Their locational properties are described by the following object definition:

#### REGION-LOCATIONAL

- AREA:
- BOUNDARY:
- SHAPE:
- POINTS:
- GRID:

Several of these are similar to the attributes of curves. AREA is the number of pixels in a region. BOUNDARY is a pointer to an instantiated curve object corresponding to the closed curve surrounding the region. Note that this curve object contains the perimeter of the region. SHAPE is a pointer to a property list for such descriptions as the MAXIMUM-BOUNDING-RECTANGLE, the CONVEX-HULL, and other statistics describing the distributions of pixels in the region or the fit of geometric shapes to the region. POINTS is the list of the coordinates of points in the region. This would not be used frequently, but is useful when there are a great many calculations requiring the points: otherwise the computation can refer back to the label associated with the region in the GRID, which is similar to that used for curves. REGIONS inherit all the properties of ISDB-OBJECTS.

As with curves, we have attributes computed for REGIONS over various images. These are stored in the REGION-ATTRIBUTES:

## REGION-ATTRIBUTES

- CONTRAST IMAGE:
- AVERAGE CONTRAST:
- CONTRAST VARIANCE:
- INTENSITY IMAGE:
- AVERAGE INTENSITY:
- INTENSITY VARIANCE:
- GENERAL-REGION-ATTRIBUTE-LIST:

There are several attributes that can be associated with regions based upon the values of it's points, such as feature histograms and statistics over various texture measures.

POINTS are treated similarly. The POINT-LOCATIONAL is:

## POINT-LOCATIONAL

- X:
- Y:
- GRID:

## POINT-ATTRIBUTE

- VALUE:

In general, point attributes are extracted directly from the image in which they occur for reasons of efficiency.

### 3.1.3 Relations

There are several types of spatial relations between image structures, describing such things as adjacency, containment, intersection, and so forth. We treat such relationships as objects which are instantiated in the ISDB. ADJACENT is described as :

## ADJACENT

- ITEM1:
- ITEM2:

Where ITEM1 and ITEM2 refer to the particular objects which are adjacent. There are specializations of the ADJACENT relationship. For example, adjacency between regions requires specification of the boundary between the regions (which is described by a curve):

## REGION-ADJACENT

- (ADJACENT)
- ADJACENCY-BOUNDARY:

Where ADJACENCY-BOUNDARY is a pointer to a CURVE. Binary relationships with special attributes handled in a similar manner:

## INTERSECT

- ITEM1:
- ITEM2:

## RELATIVE-ORIENTATION

- ITEM1:
- ITEM2:
- VALUE:

In general, the ISDB contains the results of processing and measurements, not interpretations which are expressed as hypotheses in the hypothesis data base. Thus, the ascription of the relationship of PARALLEL or ALIGNMENT

between two objects would be a hypothesis, while the measurement upon which of these hypotheses are based would be stored in an instantiated relationship for RELATIVE-ORIENTATION in the ISDB.

In general, all possible relationships between objects are not determined as objects are instantiated, but will result from computations resulting from specific queries. There are some exceptions to this due to efficiency. All region adjacencies, for example, can be computed in single pass procedure.

### **3.1.4 Non-Image Objects**

As already indicated, there are several types of objects which are not image specific, such as tables, histograms, groups, and different types of shape decompositions. Groups are selected sets of objects, such as points, lines, or regions, and are described as:

#### **GROUP**

- GROUP-TYPE:
- GROUP-CRITERIA:
- GROUP-ELEMENTS:

Where GROUP-TYPE: describes the types of entities in the GROUP; GROUP-CRITERIA describes on what basis they were selected and from what GROUP or IMAGE they were selected from; GROUP-ELEMENTS points to a list of the elements in the GROUP. A one-dimensional histogram is described as:

#### **1D-HISTOGRAM**

- PRODUCED-FROM:
- MIN:
- MAX:
- BUCKET-NUMBER:

- EXTRACTED-CLUSTERS:
- HISTOGRAM-ARRAY
- BUCKET-WIDTH

### 3.2 PROCESSING RELATIONSHIP STRUCTURE

The Processing Relationship Structure (PRS) stores information describing the processing relationships between the different objects in the ISDB and the instantiated hypothesis and tasks which invoked their creation. The PRS is a graph in which nodes store information about invoked procedures. The PRS-NODE is an object with the following attributes:

#### PRS-NODE

- PROCEDURE:
- PARAMETERS:
- ASSOCIATED-HYPOTHESIS:
- APPLIED-TO:
- RESULTS:
- ATTRIBUTES:

**PROCEDURE:** indicates the procedure that was used. The segmentation processing module contains a library of procedures for such things as particular edge operations, region extraction and shape description. These are referred to here. **PARAMETERS:** describes the parameters used in the procedures. These are such things as the number of iterations of a smoothing procedure and certain thresholds used in particular edge operators. **ASSOCIATED-HYPOTHESIS:** points to the associated hypothesis or task which invoked the procedure. A procedure is done for some reason, under the control of a perceptual grouping rule or strategy, or SAR Object Knowledge Format which has been instantiated. This is useful for reasoning about why something was done and for keeping track of context. **APPLIED-TO** describes the set of objects in the ISDB that the procedure was applied to and **RESULTS** describes the set of objects which the procedure produced. **ATTRIBUTES:** is a general property list for storing any parameters or

results which are not explicit objects in the data base.

Since all objects in the ISDB can occur in the PRS, we have a very basic object type to describe relationships in the PRS, the ISDB-OBJECT. The attributes of the ISDB-OBJECT are common to all objects:

#### ISDB-OBJECT

- PRODUCED-FROM:
- INPUT-TO:
- ASSOCIATED-HYPOTHESIS:
- ASSOCIATED-RELATIONS:

PRODUCED-FROM: describes the procedure which produced or updated a particular object. It is a list of pointers to instantiated PRS-NODES. INPUT-TO: is a list of all the procedures to which the object was used as input.

For example, consider a segmentation rule for determining whether there is a grid like structure in an image. Such a rule acts to extract globally significant structure which provides a context for directing and constraining further processing. This rule can be paraphrased as:

#### <GLOBAL-GRID-RULE>

- To determine the presence of a grid structure:
  - 1) Find long straight edges in an image
  - 2) Form a Histogram based upon edge orientation
  - 3) Extract Histogram Clusters
  - 4) Find Clusters corresponding to roughly orthogonal orientations
  - 5) Apply Evaluation Criteria to evaluate rule success

When this rule is interpreted, it produces several intermediate results. One reason for maintaining all these processing relations explicitly is so that processing steps need not be repeated. There will be several other rules based upon the set of long, straight high frequency curves. The example PRS and objects produced by execution of this rule are shown in Figures 3-2 through 3-7. The resulting internal data structure is shown in Figure 3-8.

The PRS will be useful when the system is used interactively. It will be possible to regenerate interesting results without becoming inundated in pointers to objects by backchaining from an object through the lattice of PRS-nodes.

### 3.3 QUERIES

Queries over the structures in the ISDB are implemented either as methods associated with the defined object types or as functions. We now give some examples of what such queries and methods are like. A basic function is to select from a group of entities, those with particular attributes. A method for this, defined over groups with numerical valued attributes, is:

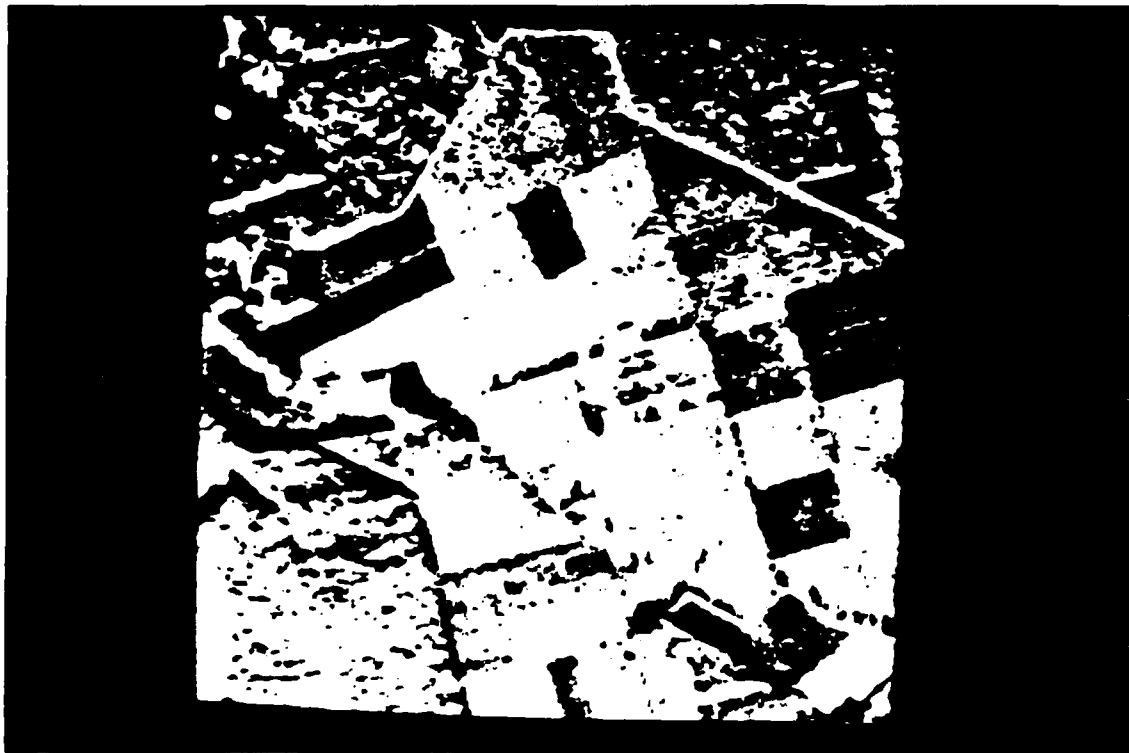
```
(defmethod (group :select-on-attributes) (attribute low high)

  (let* ((element-list (send self ':group-elements))
        (new-group (make-instance-group))
        (new-prs-node (make-instance-prs-node)))

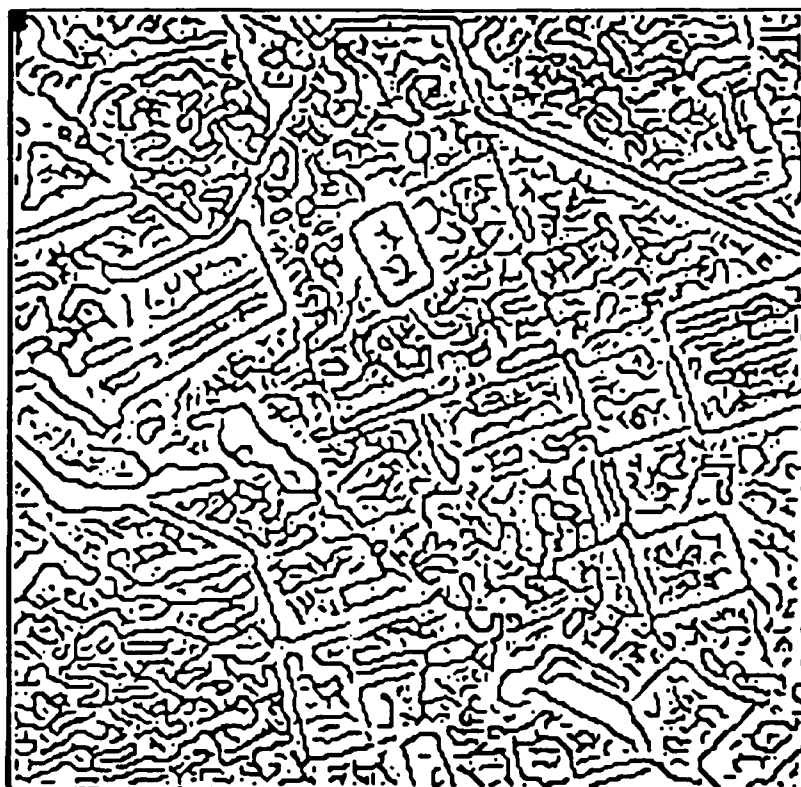
    (send new-prs-node ':set-procedure "select-on-attribute")
    (send new-prs-node ':set-parameters
      (list "attribute" attribute
            "low"      low
            "high"     high))
    (send new-prs ':set-applied-to self)
    (send new-prs ':set-results new-group)
    (send new-group ':set-group-list

      (loop for e in element-list list
        (cond ((and(>(send e attribute) low)
                  (<(send e attribute) high) )e))))))
```

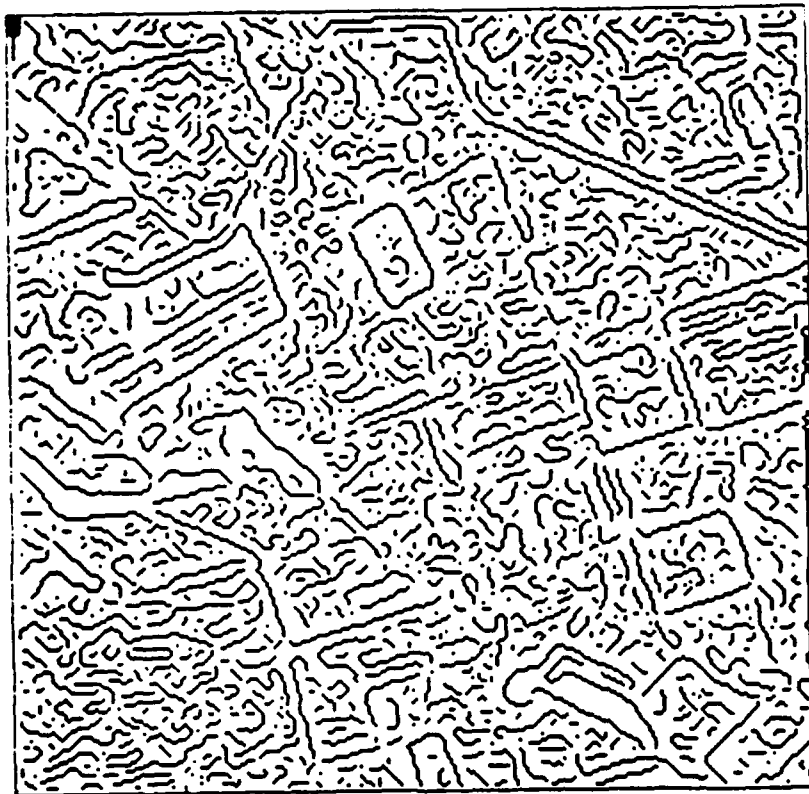
This defines a method applied to groups which will form subgroups using the criteria that the specified attribute is between low and high for the subgroup elements. The let\* statement creates a new instance of a group and an associated PRS-NODE. The attributes of each are set in a series of SENDs. Processing consists of looping through the list of objects in the group and seeing which are in the specified bounds.



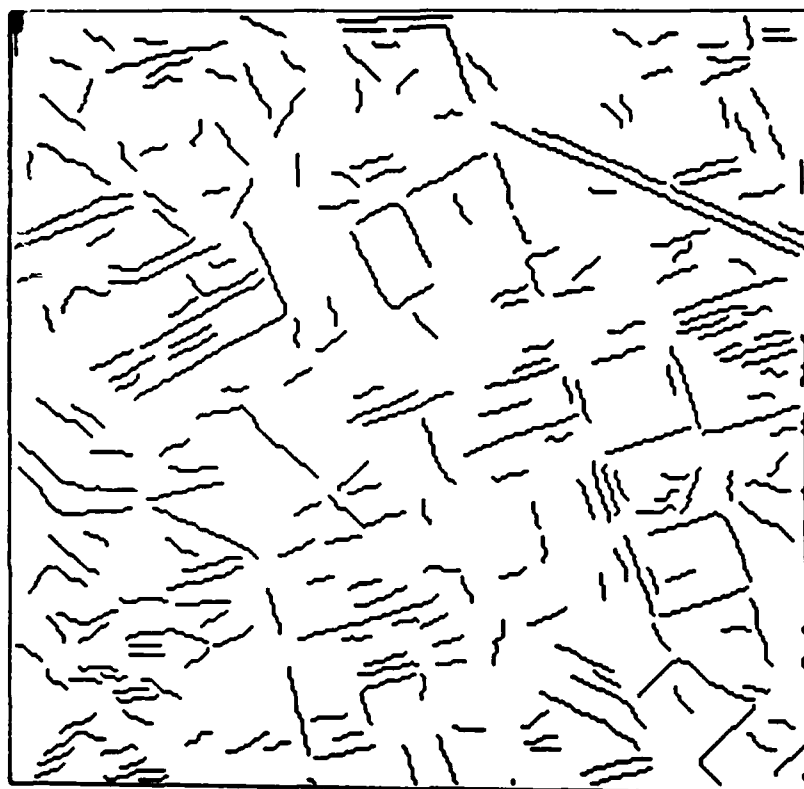
**Figure 3-2: Field Image: (ETL36)**



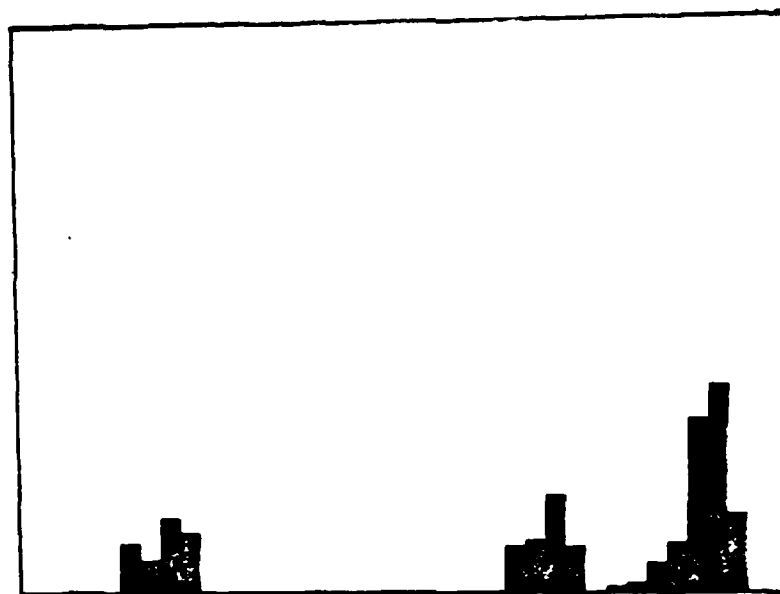
**Figure 3-3: Edge Outputs**



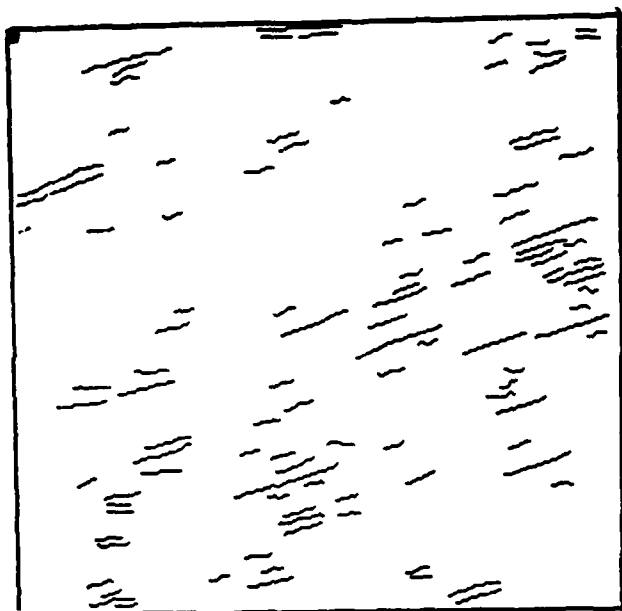
**Figure 3-4: Thinned and Labeled Edges**



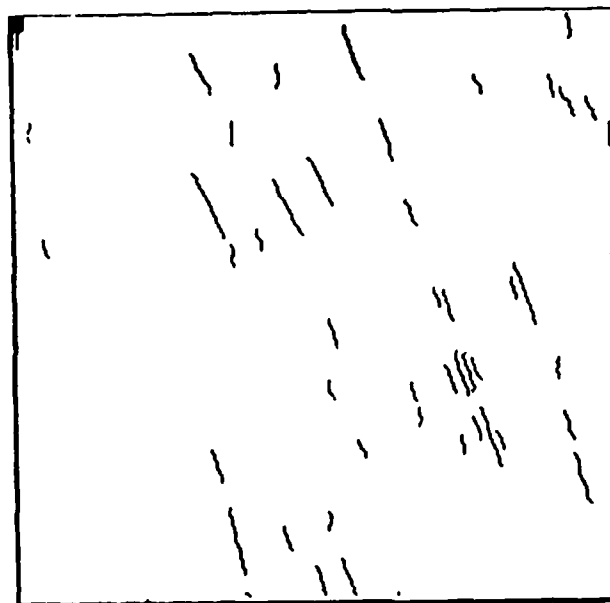
**Figure 3-5: Long Linear Subsegments**



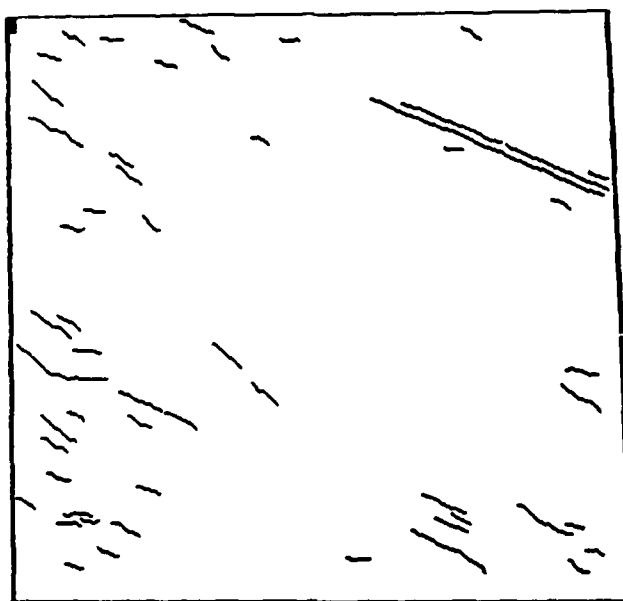
**Figure 3-6: Orientation Histogram over Long Linear Subsegments**



**Peak 1**



**Peak 2**



**Peak 3**

**Figure 3-7: Extracted Peaks Mapped onto  
Corresponding Image Structures**

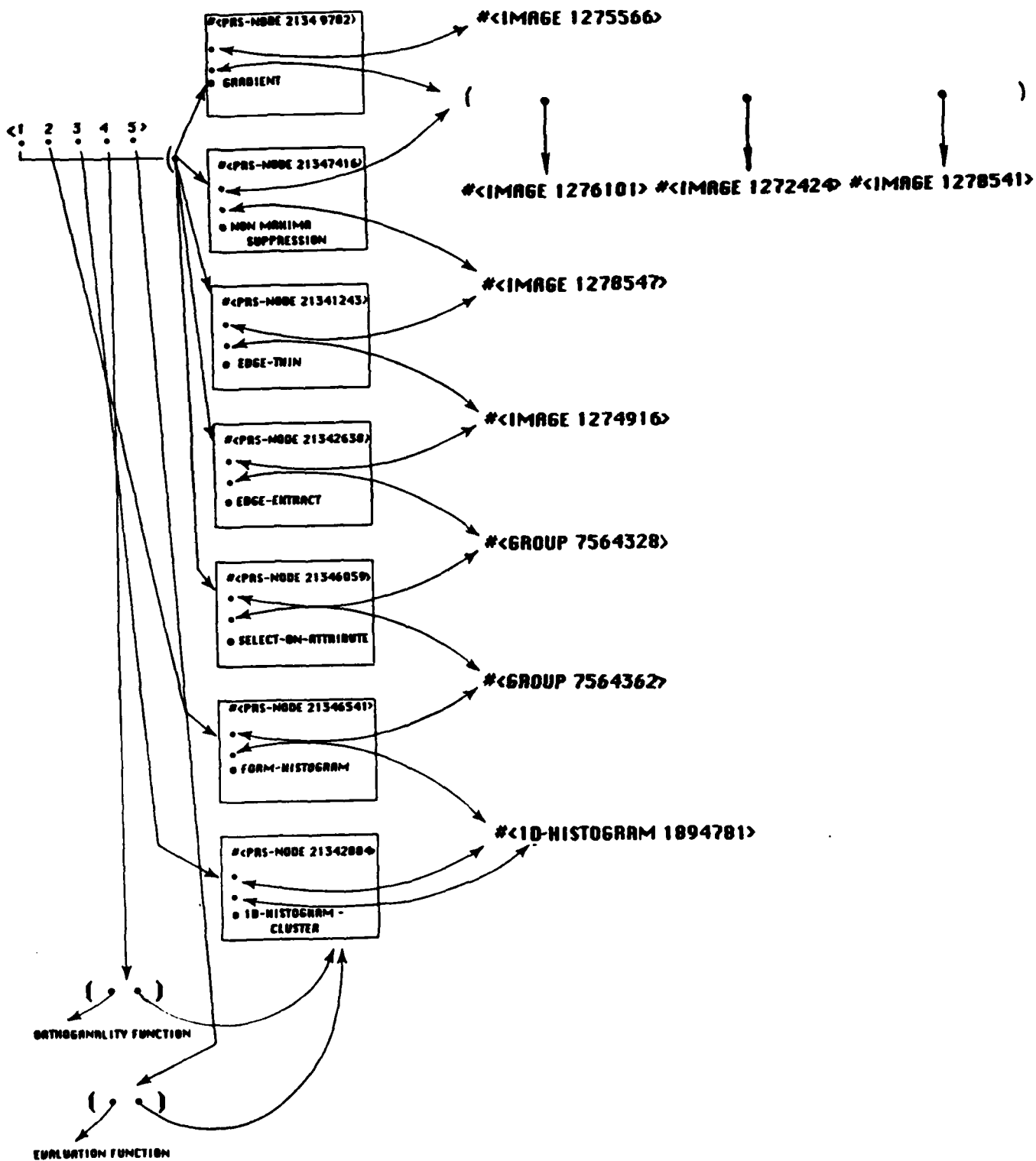


Figure 3-8: Processing Relationship Structure

To determine the histogram of some set of entities:

```
(defun object-list-histogram
  (object-list attribute min max number-of-buckets)

  (let* ((histogram (create-histogram
                        'min min
                        'max max
                        'number-of-buckets number-of-buckets
                        'proceedure "object-list-histogram"
                        'applied-to object-list)))

    (loop for e in object-list do
      (let* ((value (send e attribute)))
        (cond ((> value max) nil)
              ((< value min) nil)
              (t (send histogram ':bucket-fill value))))))
```

Here we begin to see the general style of programming supported by FLAVORS. We have a routine that will compute a histogram over any set of objects with any numerical attribute (this attribute may itself be a method which returns a number). The create-histogram statement creates a histogram and initializes the related PRS-NODE. From these specified attributes, the instantiation methods associated with the histogram object definition will determine the others, such as bucket-width. The loop statement goes through the object-list and sends the particular values to the instantiated histogram, which updates the associated buckets. Bucket-fill is a method for placing values into the histogram array.

```
(defmethod (histogram :bucket-fill) (value)

  (let* ((bucket (round (quotient (- value (send self ':min ))
                                   (send self ':bucket-width))))
        (hist-array (send-self 'histogram-array)))

    (aset (add1 (aref hist-array bucket)) hist-array bucket)))
```

To determine the set of objects within some distance of a point:

```

(defun labels-near-point (point label-image square-radius)
  (let* ((label (send label-image ':Array))
         (label-list nil)
         (dimensions (send label-image ':Dimensions))
         (max-x (sub1 (first dimensions)))
         (max-y (sub1 (second dimensions)))
         (px (first point))
         (py (second point)))

    (loop for x from (- square-radius) to square-radius do
      (loop for y from (- square-radius) to square-radius do
        (let* ((tx (+ x px)) (ty (+ y py)))
          (cond ((and (between tx 0 max-x) (between ty 0 max-y))
                 (let* ((tl (aref label tx ty)))
                   (cond ((null tl) nil)
                         ((memq tl label-list) nil)
                         (t (nconc label-list (list tl)))))))
              label-list)))

```

This is a function which takes in the point location, the object-label image to look through, and the size area to send through. The bindings in the outermost Let\* statement access the image array and its dimensions. To determine the set of objects within a masked area of an image containing pointers to ISDB objects we use the following function:

```

(defun label-select-mask (label-image mask-image)
  (let* ((label (send label-image ':Array))
         (mask (send mask-image ':Array))
         (dimensions (send label-image ':Dimensions))
         (xmax (sub1 (first dimensions)))
         (ymax (sub1 (second dimensions)))
         (object-label-list nil))

    (loop for x from 0 to xmax do
      (loop for y from 0 to ymax do
        (let ((lab (aref label x y)))
          (cond ((not (= (aref mask x y) 1)) nil)
                ((null lab) nil)
                ((memq lab edge-label-list) nil)
                (t (nconc edge-label-list (list lab))))))
      (object-label-list))

```

To determine the set of objects within some distance of an edge could then be made up from a sequence of actions and similar queries. The first is to generate a mask image from the label-image containing the object which is an

enlarged version of the object:

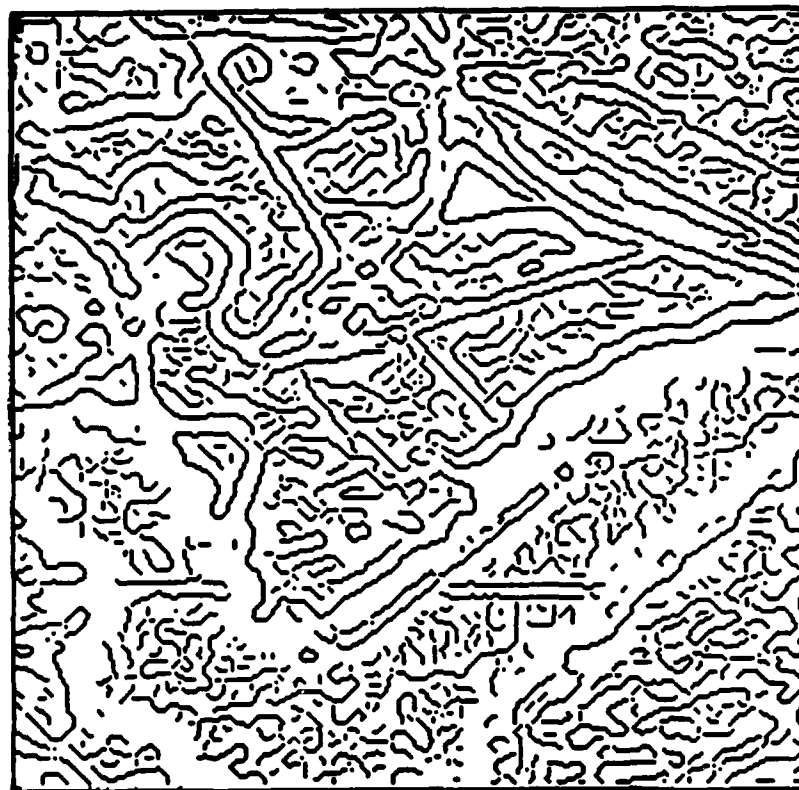
```
(defun generate-object-mask (object)
  (let* ((image (send object ':from-image))
         (image-array (send image ':array))
         (mask-image (allocate-image (send image ':dimension)
                                     art-lb
                                     (build-history
                                      (list "generate-object-mask" object)
                                      (send image ':history))
                                     (mask-image-array (send mask-image ':array))
                                     (xmax (sub1 (first (send image ':dimension))))
                                     (ymax (sub1 (second (send image ':dimension)))))))
    (loop for x from 0 to xmax do
      (loop for y from 0 to ymax do
        (cond ((eq object (aref image-array x y))
              (aset 1 mask-image-array x y))))
    mask-image))
```

The mask generated for the object is then enlarged by repeated applications of the function `fatten-mask`.

```
(defun fatten-mask (im)
  (let* ((image (send im ':array))
         (dimension (send im ':dimensions))
         (fat-mask (allocate-image dimension art-lb
                                   (build-history (list "fatten-mask" im)
                                                  (send image ':history))
                                   (loop for i from 1 to (- (nth 0 dimension) 2)
                                     do (loop for j from 1 to (- (nth 1 dimension) 2)
                                       do (cond ((= (aref image i j) 1)
                                             (loop for x from -1 to 1
                                                do (loop for y from -1 to 1
                                                  do (aset 1 thresholded-image (+ i x) (+ j y))))))))
         fat-mask))
```

The set of objects within the fattened mask is determined using the function `label-select-mask`. Note that this mask is a temporary image which could be removed using the function `deallocate-image`. There are other ways of determining the objects within some distance of a specified object using image chamfering discussed in Section 4.

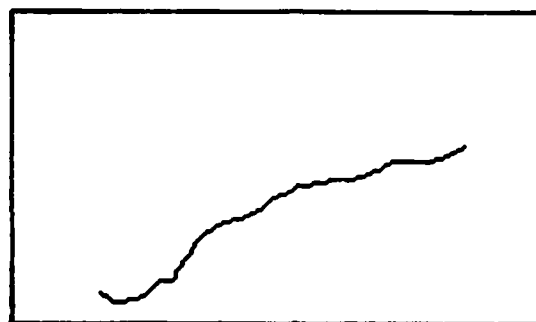
Figures 3-9 through 3-14 show these operations for such a query. Figure 3-9 shows a set of extracted curves. Figure 3-10 shows the curves selected by a select-on-attribute method to form a group of curves exceeding a minimal length threshold. Figure 3-11 shows one of these curves which is selected. Figure 3-12 shows the mask generated from this selected curve. Figure 3-13 shows the curves from the selected group which intersect the masked-area and Figure 3-14 shows those curves having most of their positions contained in the mask. Further test upon the orientation attributes of these curves could be done to determine alignment.



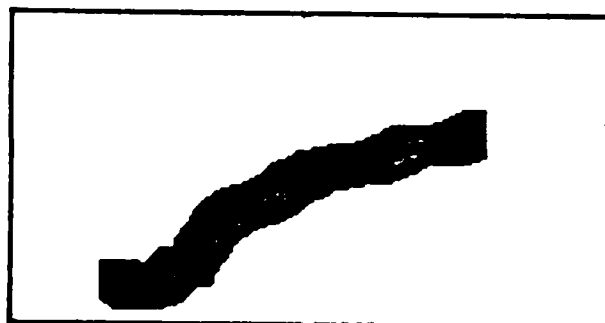
**Figure 3-9: Extracted Curves**



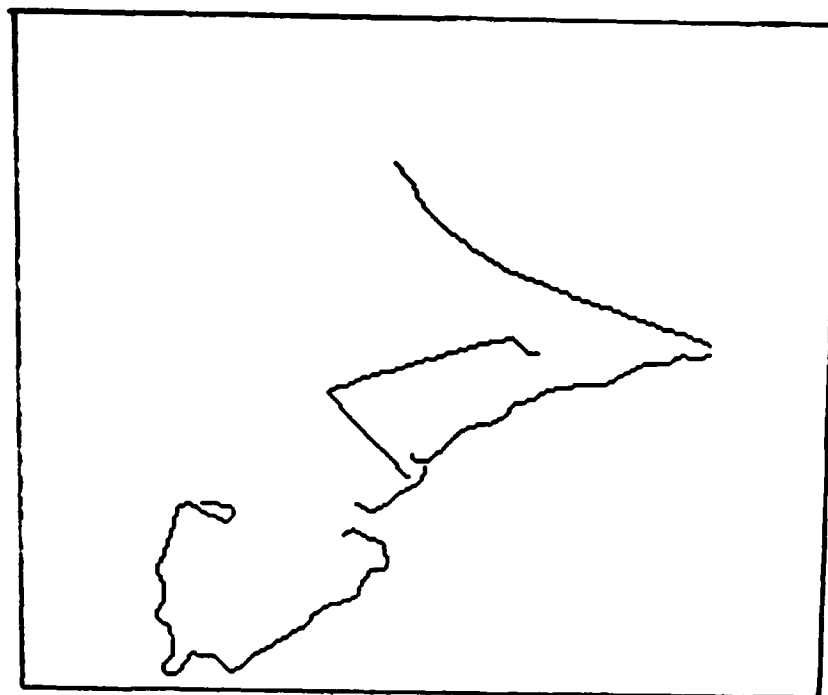
**Figure 3-10: Selected Curves Based Upon Length and Average Contrast**



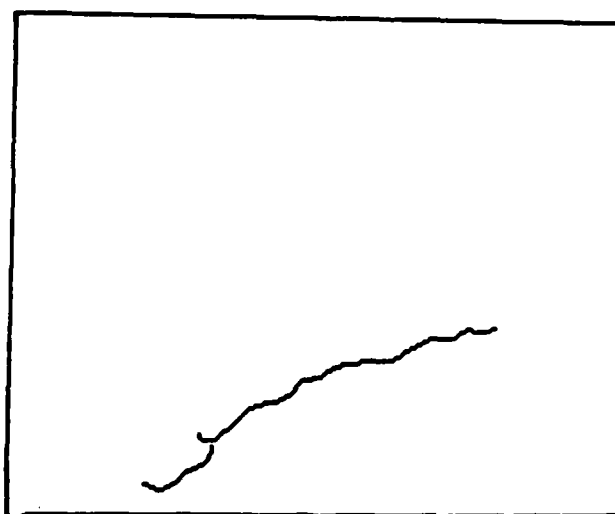
**Figure 3-11: Selected Curve**



**Figure 3-12: Mask Surrounding Curve**



**Figure 3-13: Curves Intersecting Mask**



**Figure 3-14: Curves with Dominant Intersections with Mask**

## 4. SEGMENTATION KNOWLEDGE SOURCE

This section describes the Segmentation Knowledge Source. We begin with the requirements upon segmentation procedures for incorporation into an automated vision system. The section then describes the particular edge, region, and shape extraction procedures implemented and analyzed. These serve as the basic processes that the system uses to extract image structures. Finally, the organization of segmentation knowledge in terms of rules that facilitate both model and data-driven processing is described.

### 4.1 IMAGE SEGMENTATION

Image Segmentation is concerned with breaking an image into structural components, such as regions, boundaries, edges, and points, that can be used throughout the interpretation process. There has been significant work in the last 25 years in developing such techniques. Still, this work has not resulted in automatic image interpretation systems. Primarily this is because such routines do not decompose an image into structures which correspond to world objects. World objects are semantically determined entities whose extraction requires contextual and object-specific knowledge which cannot be easily incorporated into, for example, low level filtering operations. That is, it is impossible to make a general filter that will detect roads. It is possible, however, to automate the reasoning about the segmentation procedures that can be used in the extraction of roads based upon a priori information and the status of the ongoing image interpretation process. We see automating this process of reasoning about segmentation as the basic research task we are addressing.

There are some general properties that low level vision processing must incorporate for such flexible application in automated image interpretation. First, the segmentation processes must be explicitly understood in terms of the types of image information they are sensitive to and can extract. This entails relating the parameters controlling a particular segmentation process and the kinds of image structures that will be extracted. We express this as rules relating different image properties and the parameter settings for particular segmentation procedures. A basic example of this is the use of segmentation procedures which

are selectively sensitive to different spatial resolutions such as zero-crossing extraction [Marr - 80]. A second example is being able to manipulate the cluster formation process in histogram-based segmentation based upon expected image events. In addition, the segmentation processes should be applicable, with different parameter settings, to restricted portions of an image that will be isolated for focused processing during the interpretation process. All of this allows the interpretation process to have active and intelligent control over the segmentation process itself. Intelligent segmentation requires a symbolic representation of the structures in an image and the contexts in which they were extracted. This enables the segmentation process to be based upon general relations and attributes. Segmentation in truly autonomous computer vision systems can not consist of applying standard edge and region routines to an image and then interpreting the results. It is an intelligent, problem-solving activity that requires rules and strategies over symbolic representations.

## **4.2 EDGE EXTRACTION PROCEDURES**

Edge extraction is fundamental to image processing. It involves several things: the basic operators for describing local changes in image intensity; the various grouping and thinning operations for combining these local measurements into linear features; and several different shape descriptions and approximations that can be associated with these linear features. From our perspective, we favor the use of simple edge operators whose responses are directly related in a clearly understood way to underlying image properties. They should also be tunable for selective sensitivity to particular types of image structures based upon such things as spatial frequency or specific predictions from an environmental model. We also believe that a wide range of segmentation grouping and thinning operations are necessary but that the effects and application of these requires explicit representation and should not be implicitly contained in some process. For example, the Nevatia-Babu [Nevatia - 80] edge operator is often adequate at extracting linear segments of high contrast, but it contains a wide range of parameters whose effects are not explicitly understood in terms of underlying image structure. This makes the operator very hard to model and understand and thus to apply in an intelligent manner in an automatic system. Operators for which there is an explicit model for the relation between image structure and operator response are the Canny edge operator [Canny - 83], the Marr-Hildreth operator,

Burt's pyramid operations [Burt - 81 and Burt - 82] and Haralick's topographic primal sketch and slope-facet edge models [Haralick - 81 and Haralick - 83]. These are also multiresolution operators for hierarchical processing. We now describe some of these.

#### 4.2.1 Zero-Crossing Extraction

Zero-Crossing based edge extraction was introduced by Marr and Hildreth [Marr - 80] and has since been used in a variety of applications [Grimson - 85]. Computationally, it can be expressed as a three step procedure applied to an image:

1. Convolution with a Gaussian mask to select contrast at different spatial frequencies.
2. Convolution with a Laplacian mask to determine points of significant intensity change.
3. Thresholding the result of Laplacian convolution at Zero to extract closed contours along which the image intensity changes are maximal. This corresponds to finding the zeros of a second derivative.

Zero-Crossings can be extracted by other means. The Laplacian of a Gaussian can be expressed as a single convolution mask (the Mexican hat operator related to center/surround cells in animal retinas. This can also be computed by a difference of Gaussians. The physiological reality of Zero-Crossings has been an active area of interest since Marr's work first appeared.). This convolution can be performed using the Fourier Transform or other computational speed-ups possible for symmetric convolutions [Canny - 83]. There is also current interest in performing the convolution optically [Grimson - 85].

Figure 4-1 shows a selected portion of an image of some fields. Figures 4-2 through 4-7 show the sequence of zero-crossing regions and boundaries extracted using increasingly wider Gaussians. Figure 4-2 shows the Laplacian of the raw image. Note the high frequency noise producing vertical banding and how this is

filtered out at the lower spatial frequencies. Figures 4-8 through 4-9 show the extracted contours based upon thresholding the contrast at the zero-crossing. Figure 4-10 shows the threshold zero-crossings from the different spatial frequencies in the previous figures combined if a majority of them had zero-crossings at image points.

#### **4.2.2 Canny**

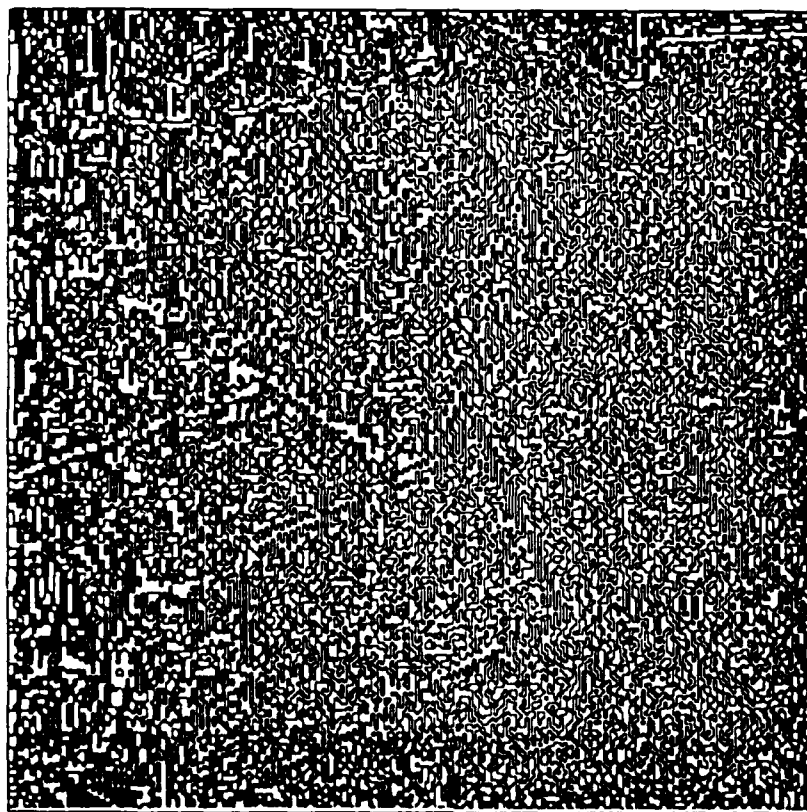
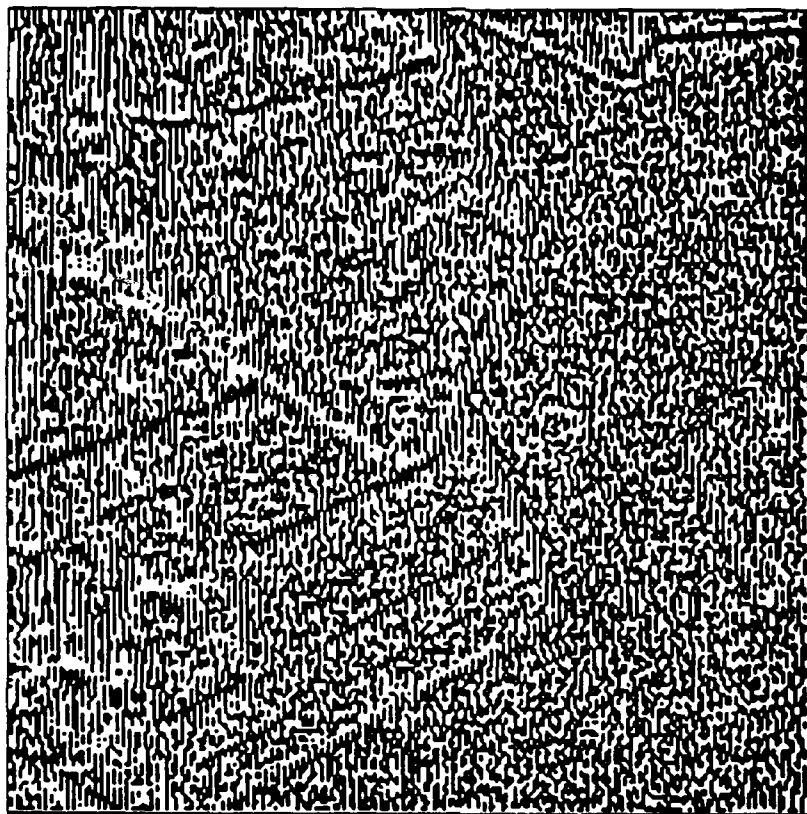
The Canny edge operator is a directional multi-resolution edge operator with many appealing properties [Canny - 83]. First, it was derived by a variational argument. That is, the operator is guaranteed to meet certain optimality conditions for a particular type of edge. In Canny's derivation, the optimality criteria were developed for position and unique detection of step edges in Gaussian noise. Second, the operator is tunable for sensitivity to edges at different spatial frequencies. And finally, the operator has many interchangeable components for such things as using masks to calculate edge support over larger areas, noise estimation, and linking local edge measurements together.

The basic steps of the operator are:

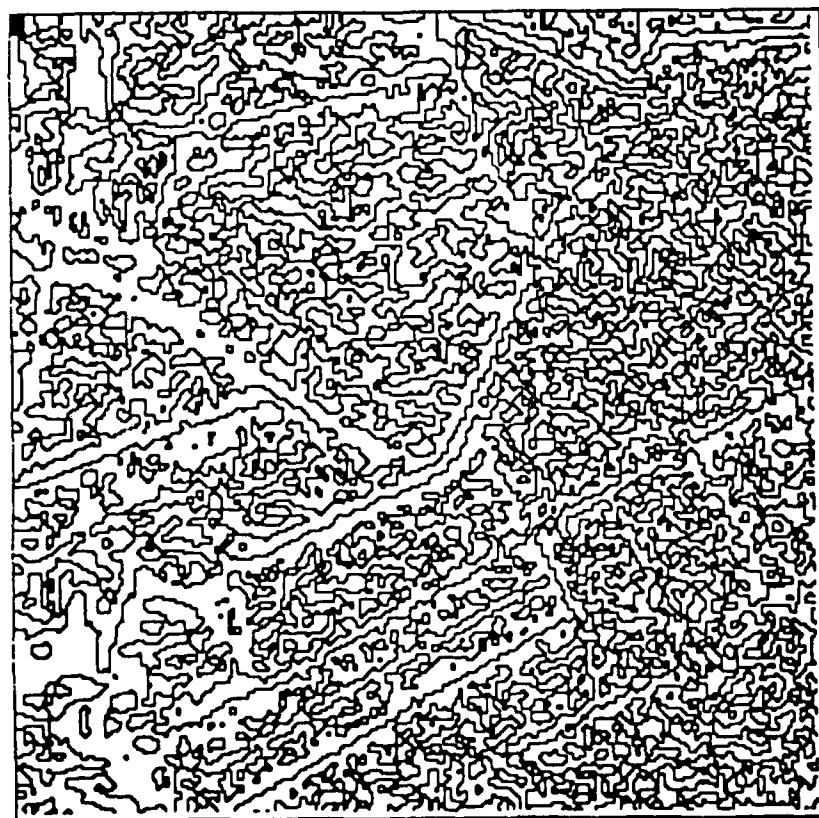
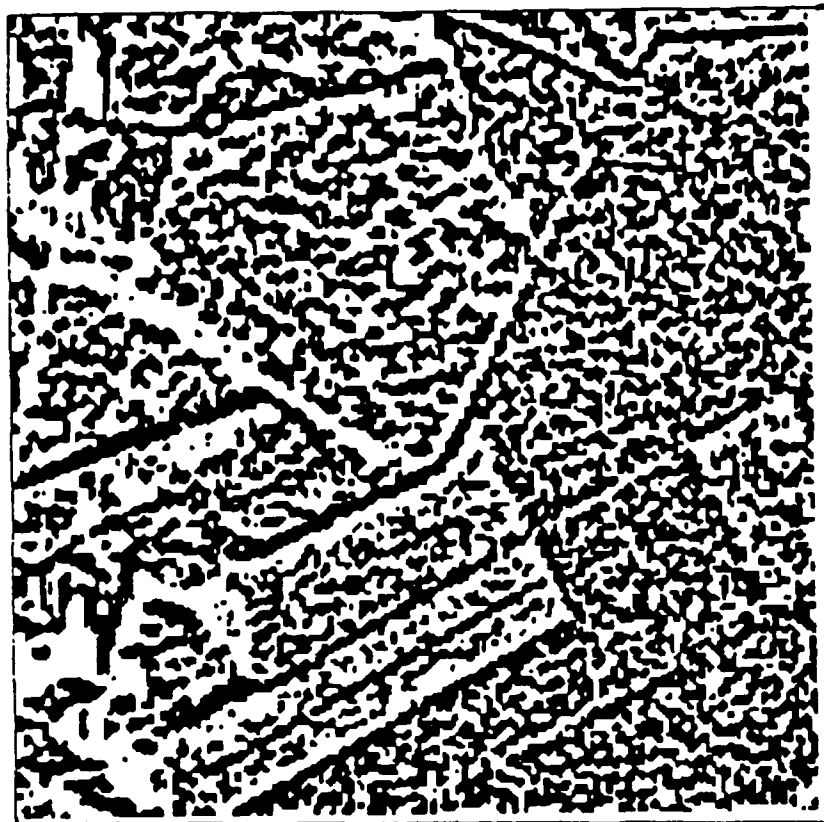
- 1) Smooth the image with a Gaussian mask, to effectively filter the desired spatial frequencies. The convolution can be done in many ways and Canny's thesis contains an excellent review of techniques for symmetric convolutions. In our work, we have used the simplest of these, based upon 1-D mask convolutions in two orthogonal directions.
- 2) Calculate the gradient of the image. This is done by centering the gradient vector on the center of the 2x2 mask in Figure 4-11 and calculating the gradient components from the difference in the two orthogonal directions.
- 3) The local maxima in the gradient magnitude are determined. This is done for a given gradient vector by interpolating the gradient in the forward and backward direction along the vector at the points indicated in Figure 4-12, projecting the interpolated gradient at these points onto the line to determine the interpolated gradient magnitude. A point is



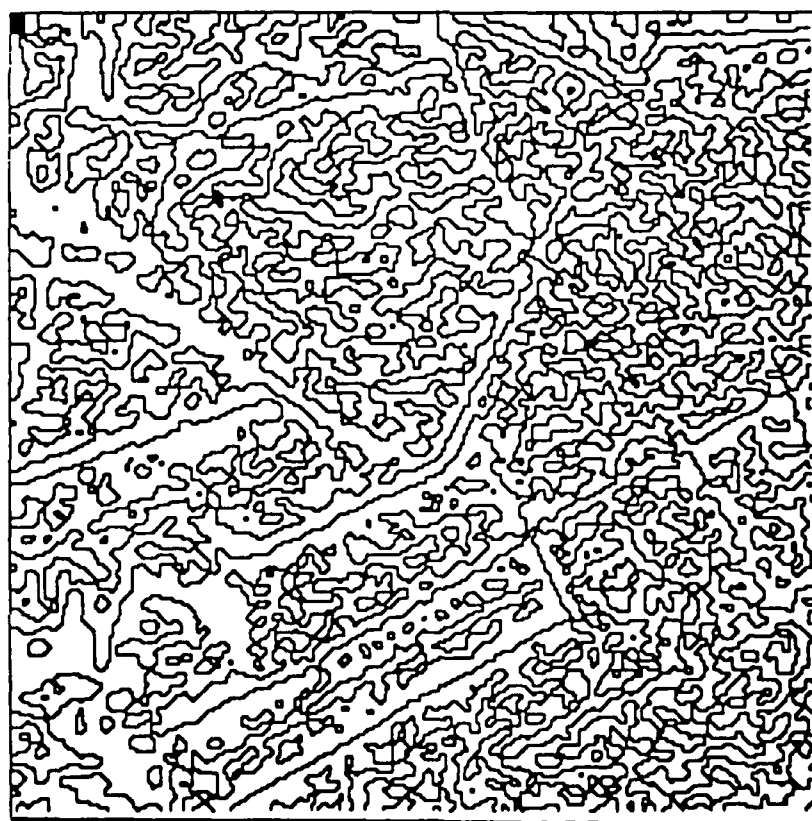
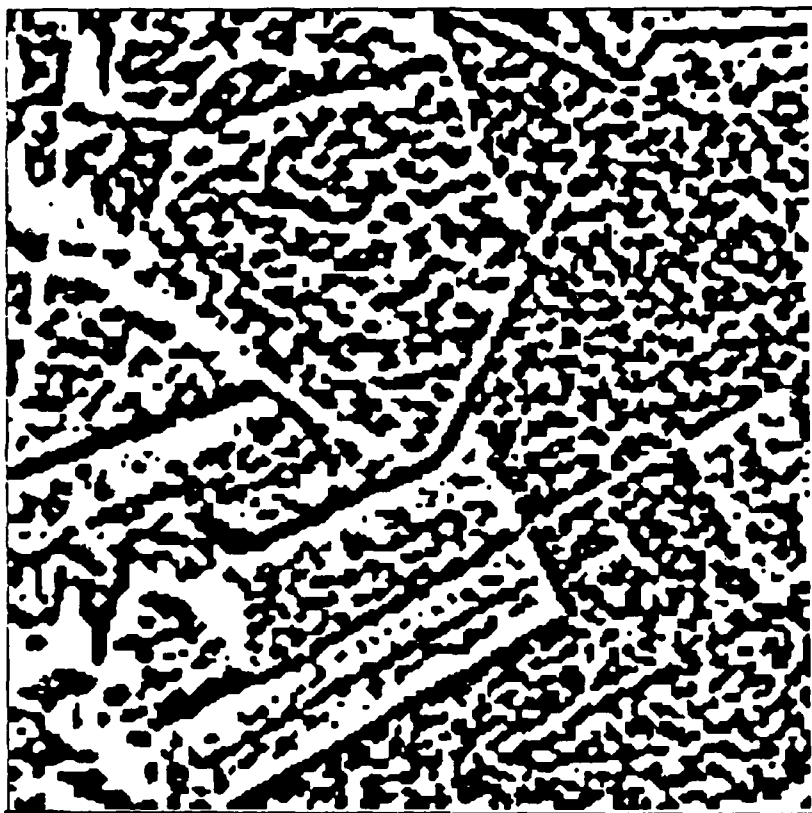
**Figure 4-1: Selected Field Image: (ETL36 Sub-Image)**



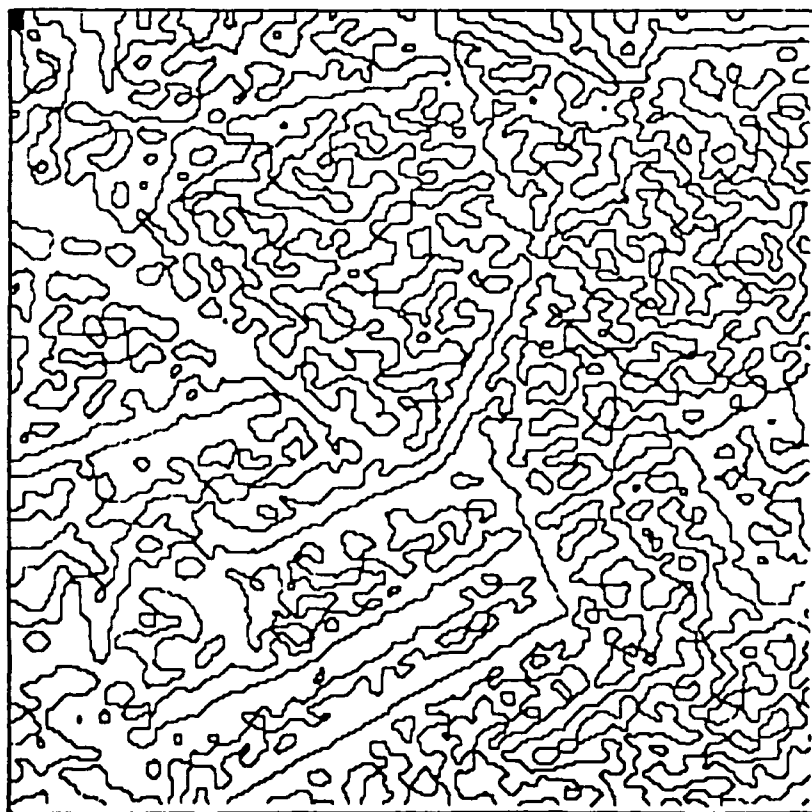
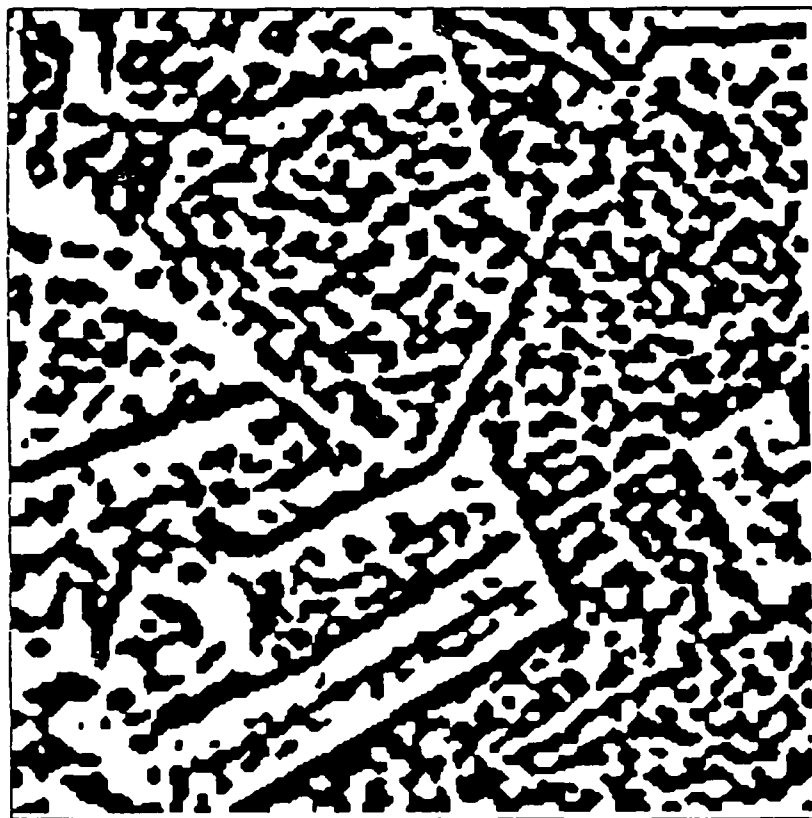
**Figure 4-2: Extracted Zero-Crossing Regions and Contours**



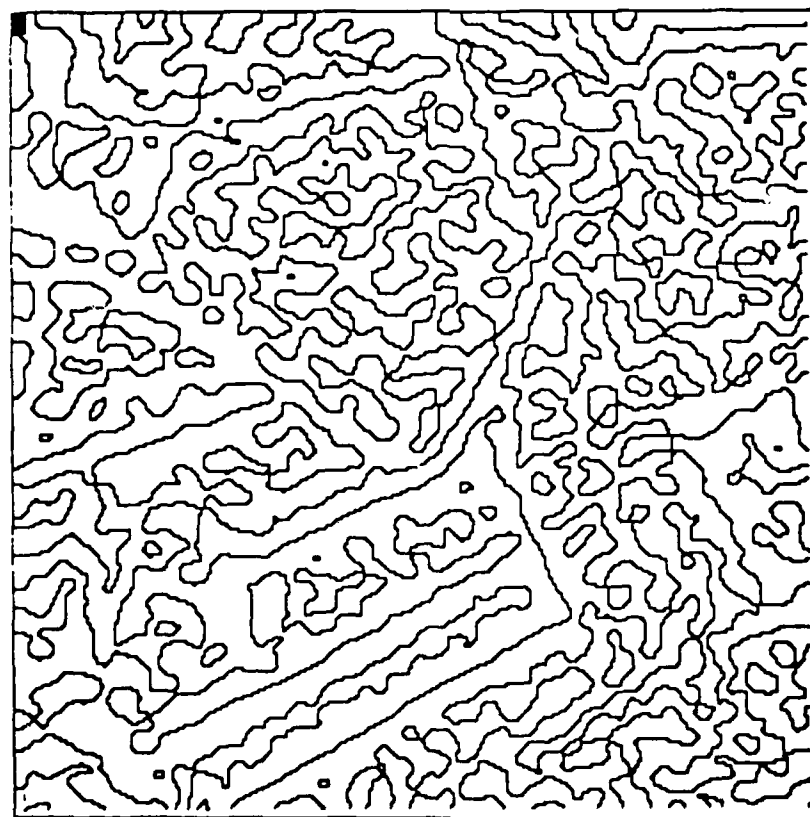
**Figure 4-3: Extracted Zero-Crossing Regions and Contours (cont'd)**



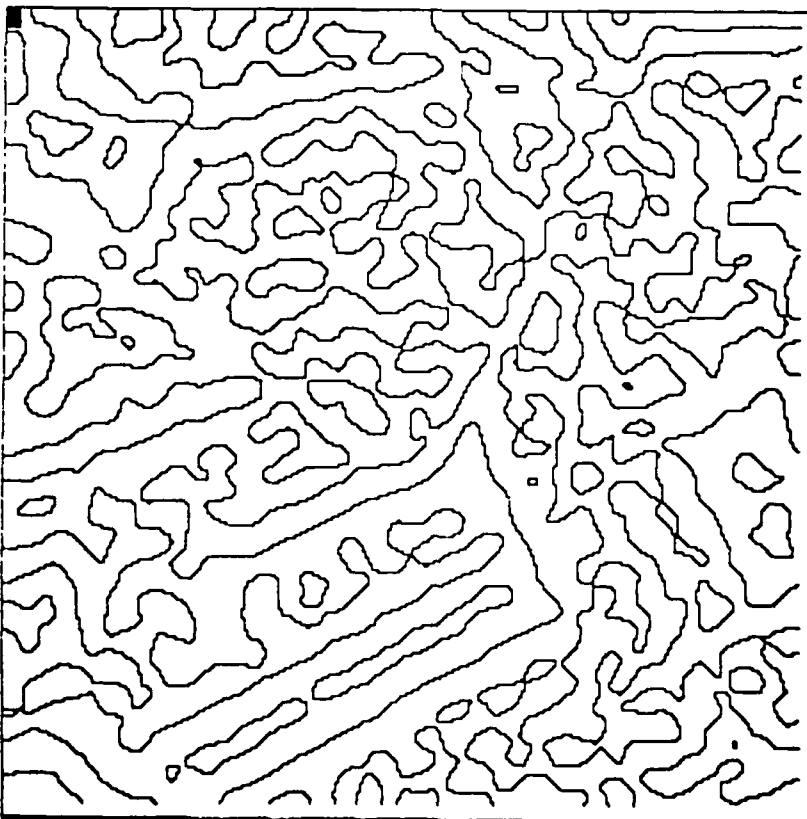
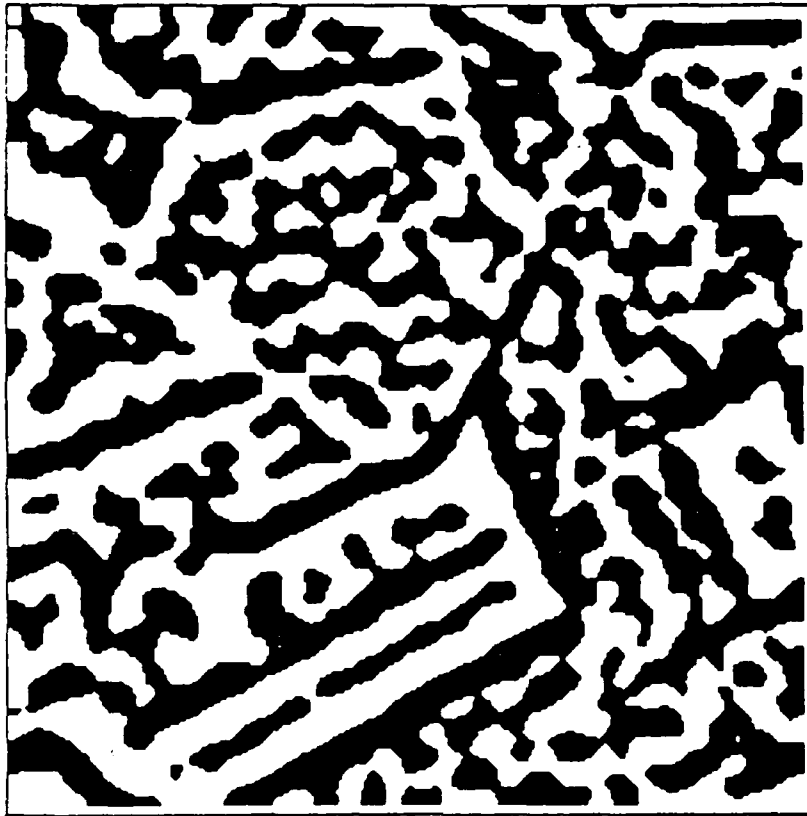
**Figure 4-4: Extracted Zero-Crossing Regions and Contours (cont'd)**



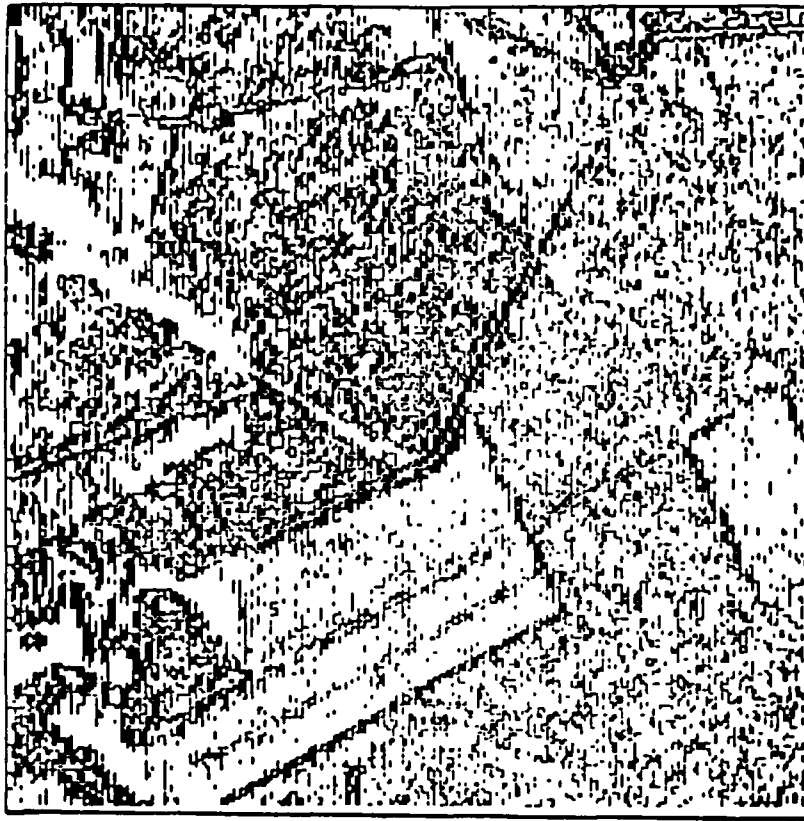
**Figure 4-5: Extracted Zero-Crossing Regions and Contours (cont'd)**



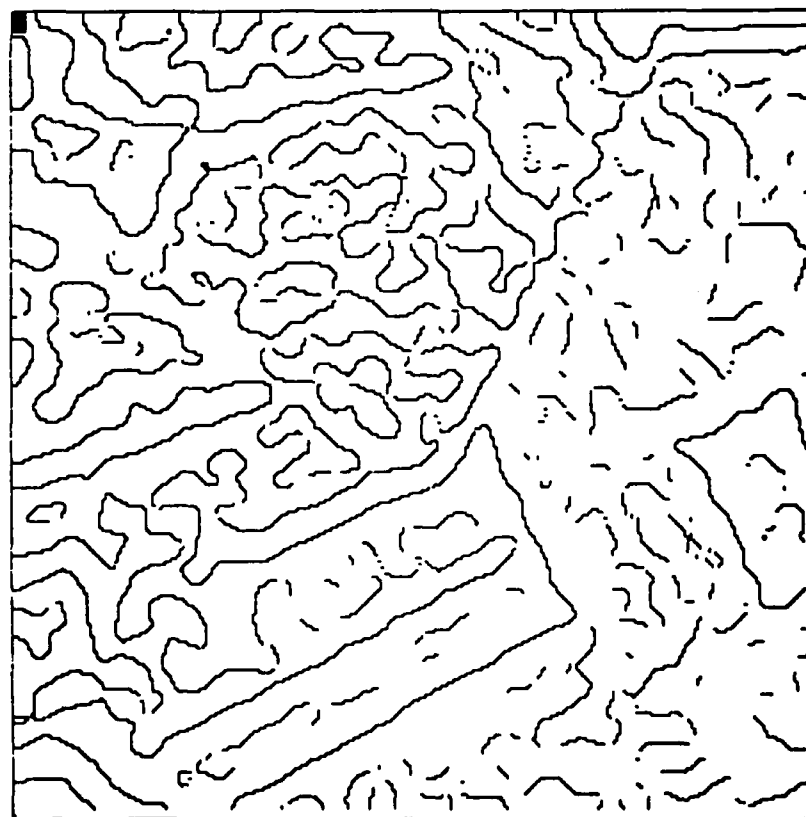
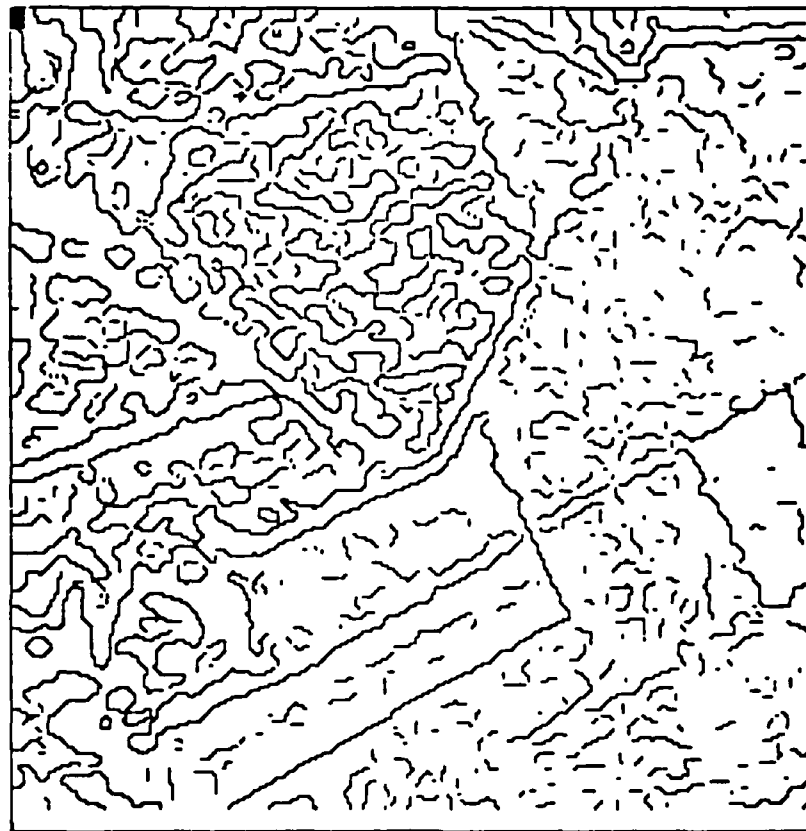
**Figure 4-6: Extracted Zero-Crossing Regions and Contours (cont'd)**



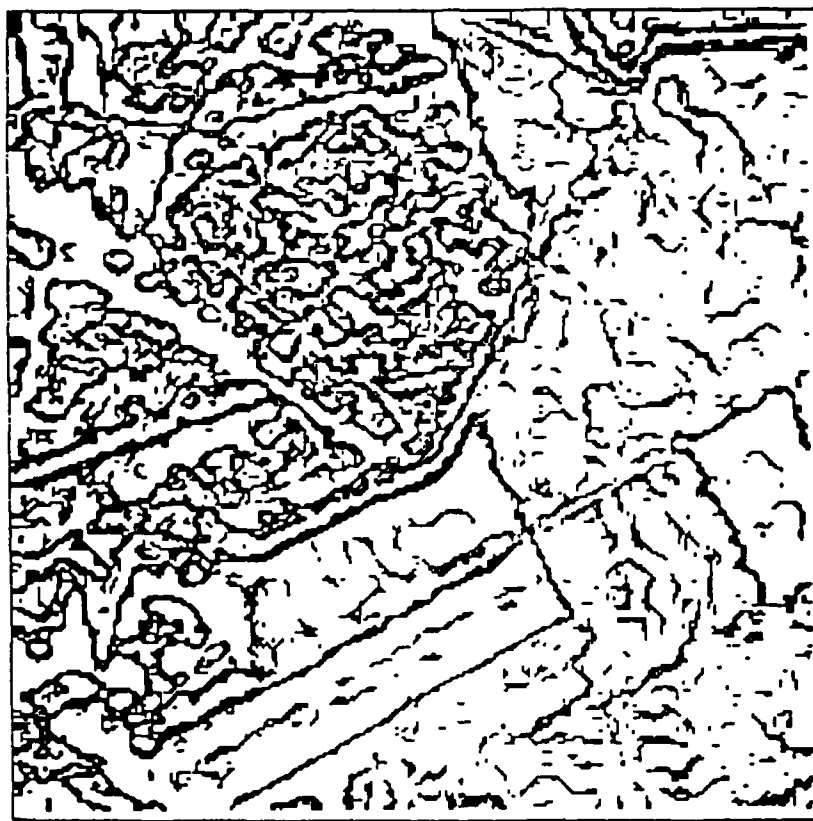
**Figure 4-7: Extracted Zero-Crossing Regions and Contours (cont'd)**



**Figure 4-8: Thresholded Zero-Crossing Contours**



**Figure 4-9: Thresholded Zero-Crossing Contours (cont'd)**



**Figure 4-10: Superimposed Zero-Crossing Contours**

tagged if it is a local maximum in gradient magnitude with respect to its interpolated neighbors.

- 4) The extracted maxima are then evaluated with respect to a threshold corresponding to the extent to which they are maximal relative to their neighbors based upon a global noise estimation.

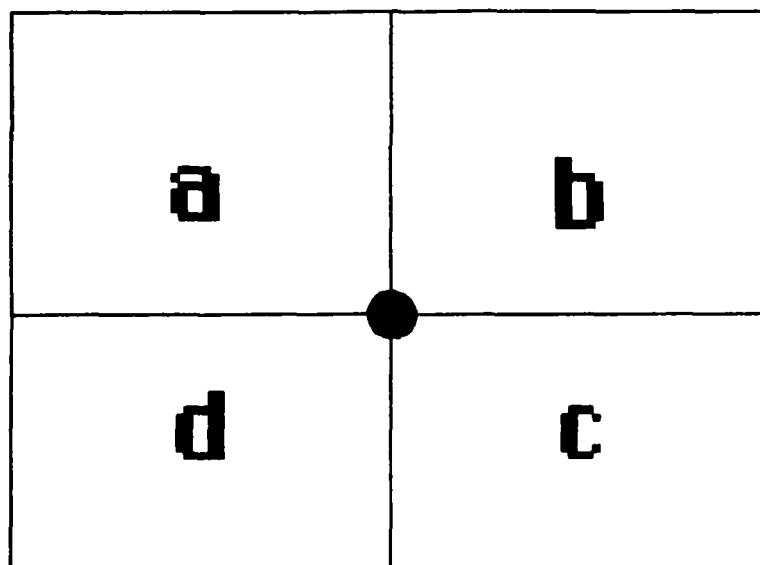
These values may then be further evaluated with respect to the values at neighboring points in a hysteresis edge linking. In our implementation, we select the edges based upon other explicit criteria concerning average attributes along the edge or shape relations with nearby edges. It is also necessary to perform an 8-connected edge thinning to obtain contours which can be traversed.

Figures 4-13 through 4-15 show the output of the Canny edge operator at different spatial frequencies corresponding to increasingly large Gaussians, applied to the image in Figure 4-1.

We have found the Canny edge operator to give very good results. It can be used in a tunable fashion and will generally pull out any observable edge. One problem, which is not particular to the Canny edge operator, but instead dealing with noisy imagery, is that it will miss low contrast, high frequency features in such images.

#### **4.2.3 Burt**

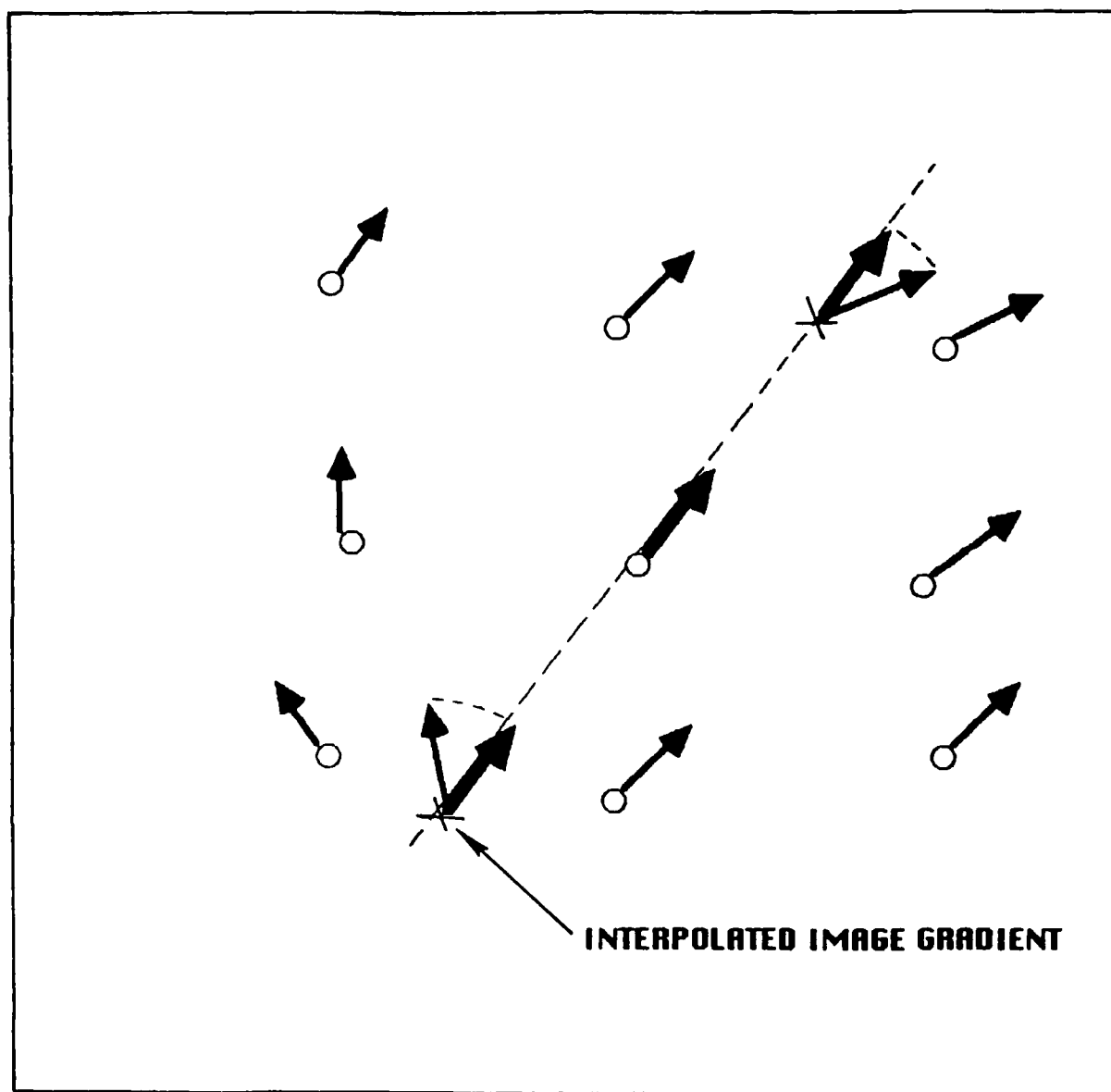
The Burtian Pyramid [Burt - 81 and Burt - 82] provides simple, fast techniques for determining image properties and representing image properties at multiple levels of resolution. The processing is based upon the formation of two different hierarchical representations of an image. The first is called the GAUSSIAN PYRAMID and is formed by smoothing an image with a 5x5 mask which approximates a Gaussian, subsampling the resulting image at every other pixel to reduce resolution and form a reduced image. This can then be applied iteratively to produce a sequence of images, each 4 times smaller than the one it was generated from. Each level of the Gaussian Pyramid corresponds to the image information at a lower spatial frequency. The reduction operation can be applied rapidly.



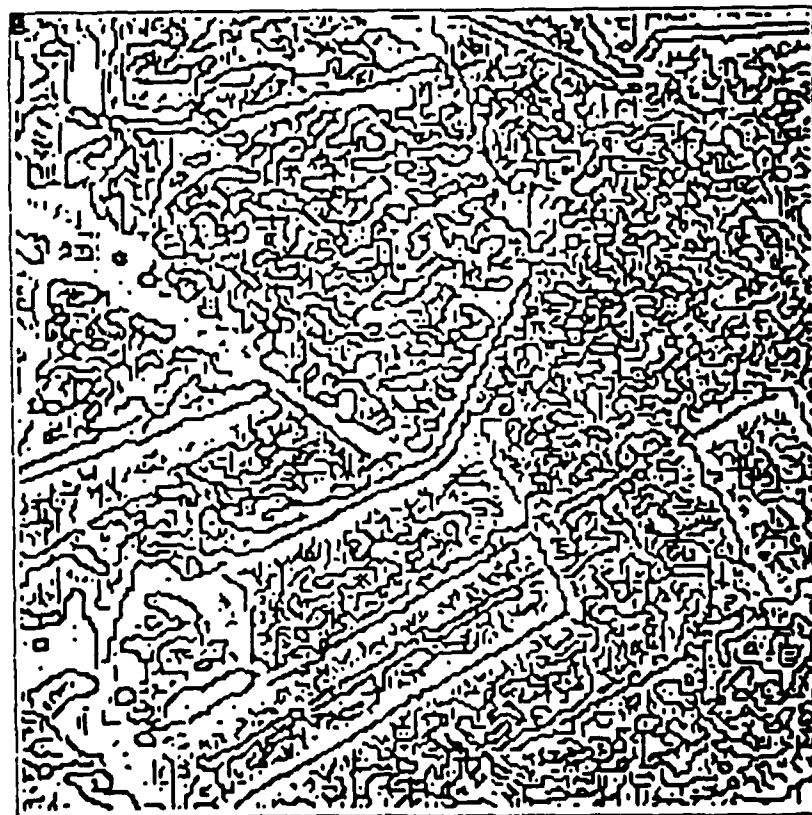
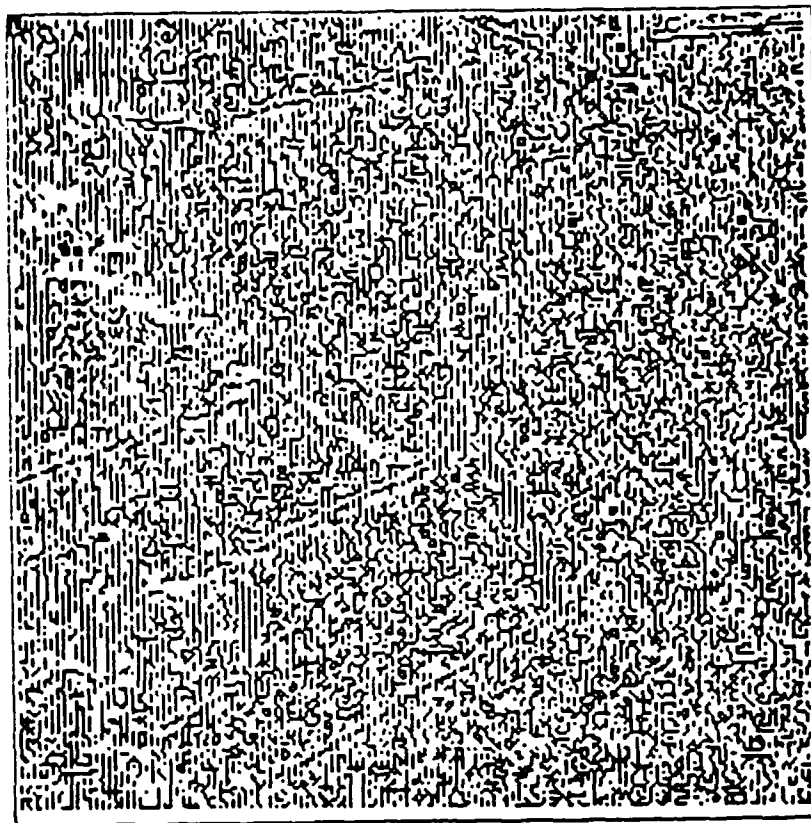
**GRADIENT AT CENTER OF 2 x 2 MASK =**

$$((a + d - b - c), (a + b - d - c))$$

**Figure 4-11: Gradient Calculation**



**Figure 4-12: Gradient Interpolation**



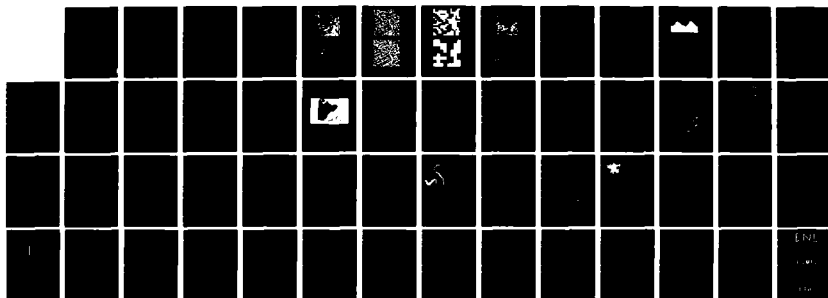
**Figure 4-13: Canny Edges**

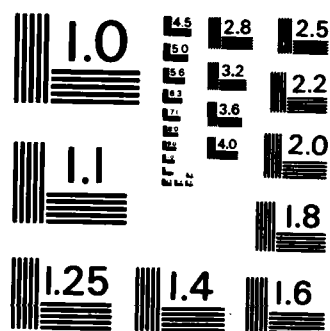
UNCLASSIFIED

DACH72-84-C-0014

2/2

ML





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS - 1963-A

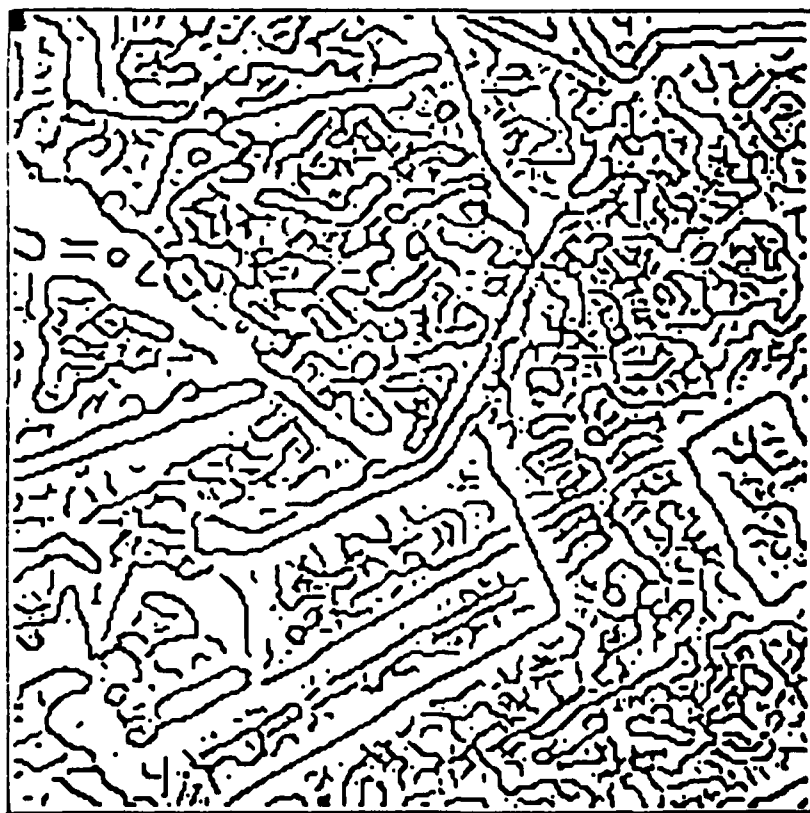
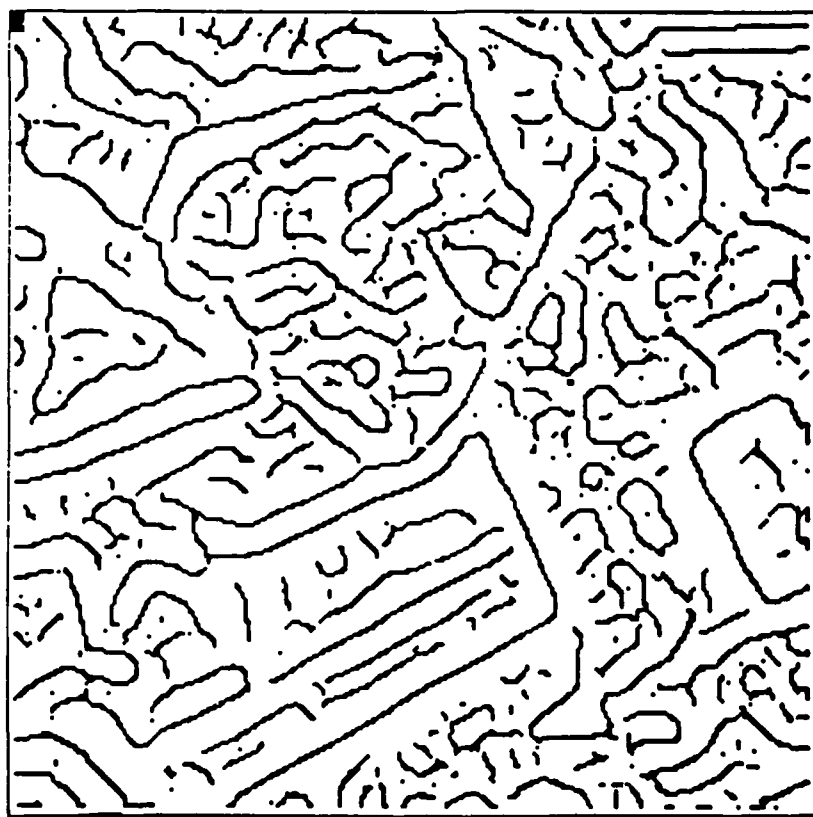
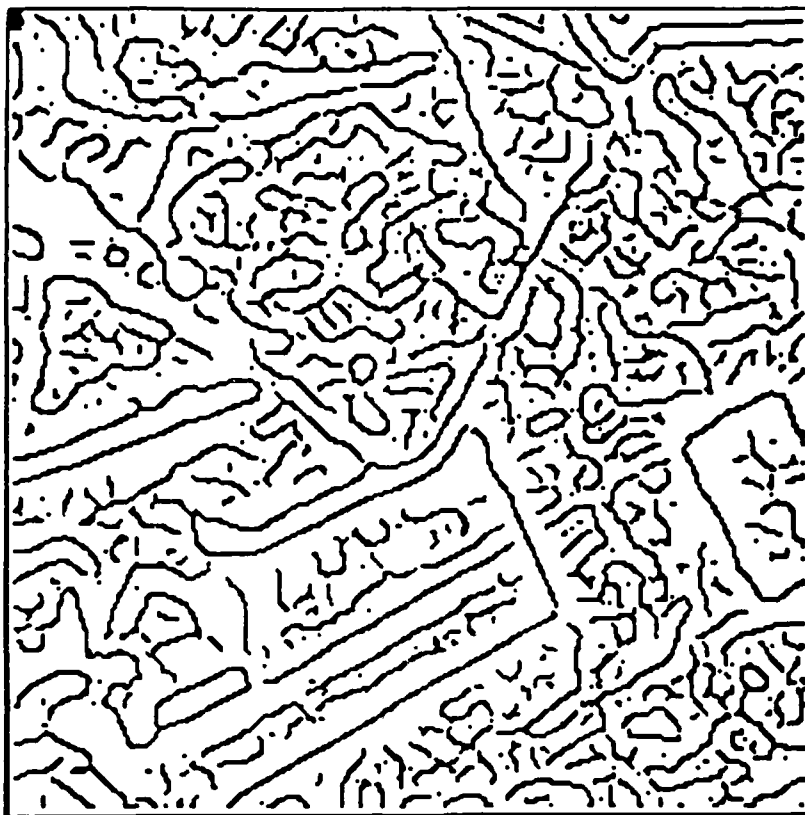


Figure 4-14: Canny Edges (cont'd)



**Figure 4-15: Canny Edges (cont'd)**

The contrast information in the Gaussian Pyramid is computed by the formation of the LAPLACIAN PYRAMID. This can be produced in two ways. In one of these, a 5x5 Laplacian operator is applied to each level of the Gaussian Pyramid to generate the corresponding level of the Laplacian Pyramid. In the other, the  $n$ th level of the Laplacian Pyramid is formed by expanding the  $n+1$  level of the Gaussian Pyramid to the same resolution as the  $n$ th level and subtracting the two. This corresponds to the fact that the difference of Gaussians will approximate zero-crossings. Thresholding at zero yields the zero-crossing contours.

Figure 4-16 shows the zero-crossing regions at the different levels of the Laplacian pyramid obtained for the image in Figure 4-1. Figures 4-17 and 4-18 show these images at a normalized resolution. Figure 4-19 shows the thresholded contours from the zero-crossings.

#### 4.2.4 Hough Transform

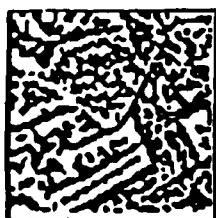
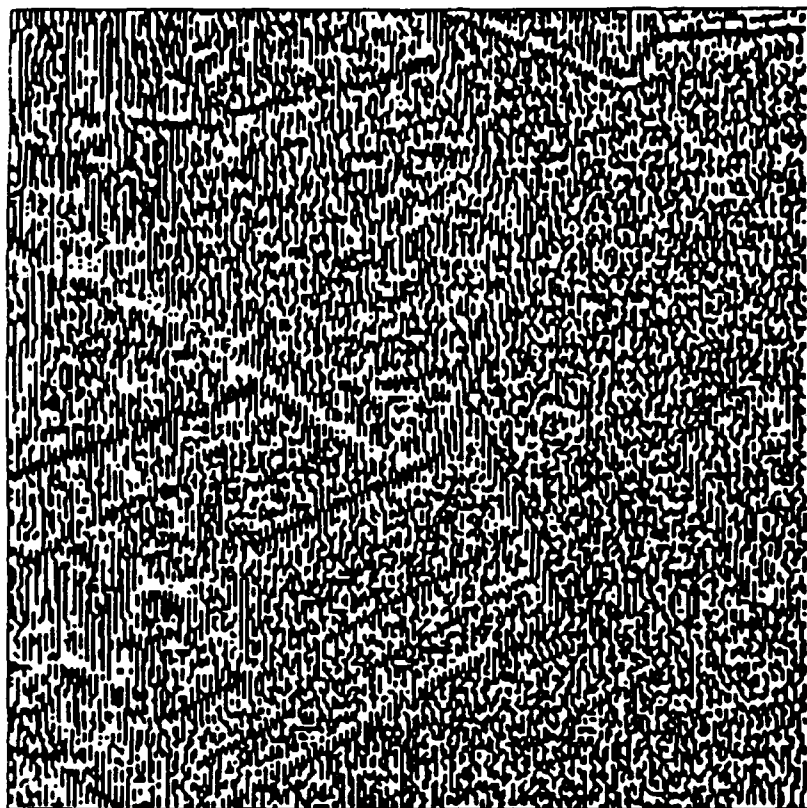
The Hough Transform [Hough - 62] is a global histogram technique for edge extraction. In it, each image point "votes" for the parameters describing the line perpendicular to the gradient at the point. Parameter buckets containing the most votes will correspond to straight line segments in the image. The Hough Transform is an effective technique of extracting and grouping spatially disconnected edge segments. More complicated curves require more parameters. The selection of bucket-size is a critical issue.

There are several different parameterizations which can be used to describe lines in the image plane. The conventional one is (Figure 4-20):

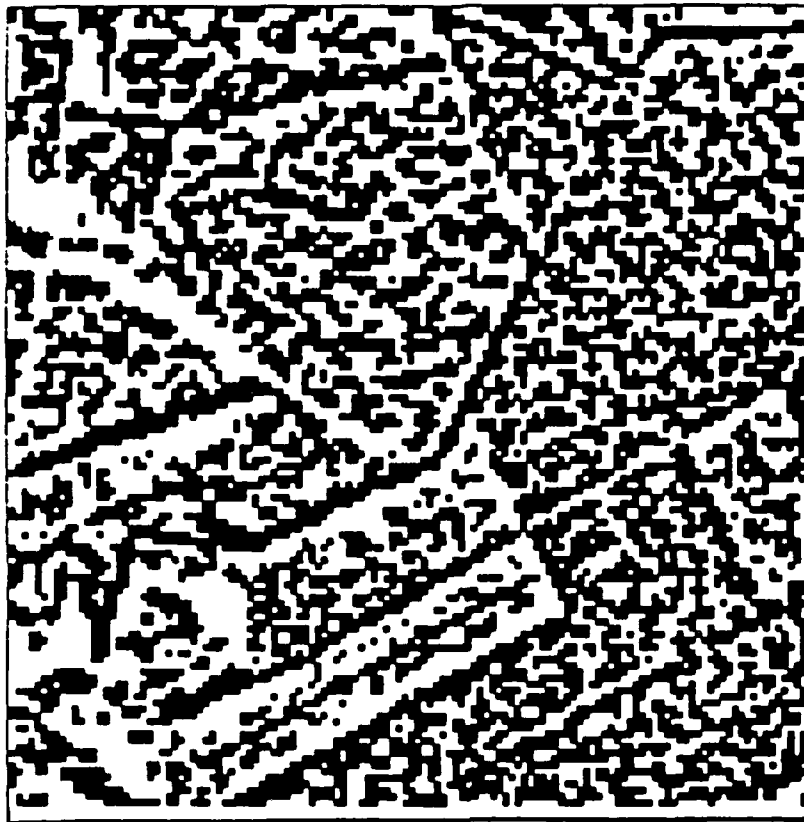
$$r = \frac{\Delta x x + \Delta y y}{\sqrt{\Delta x^2 + \Delta y^2}}$$

$$\Theta = \tan^{-1} \left( \frac{\Delta y}{\Delta x} \right)$$

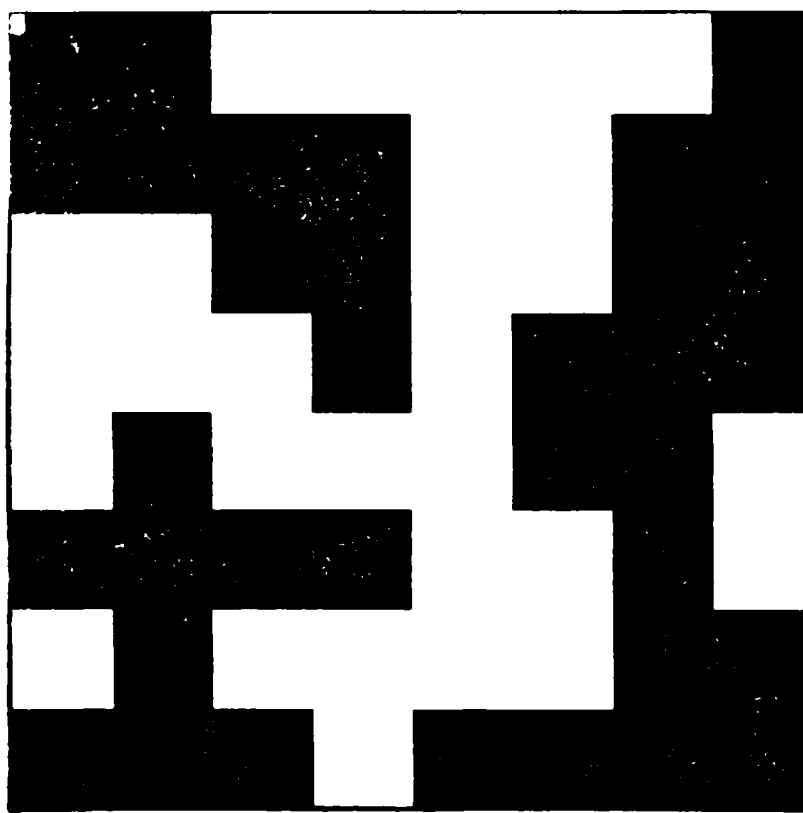
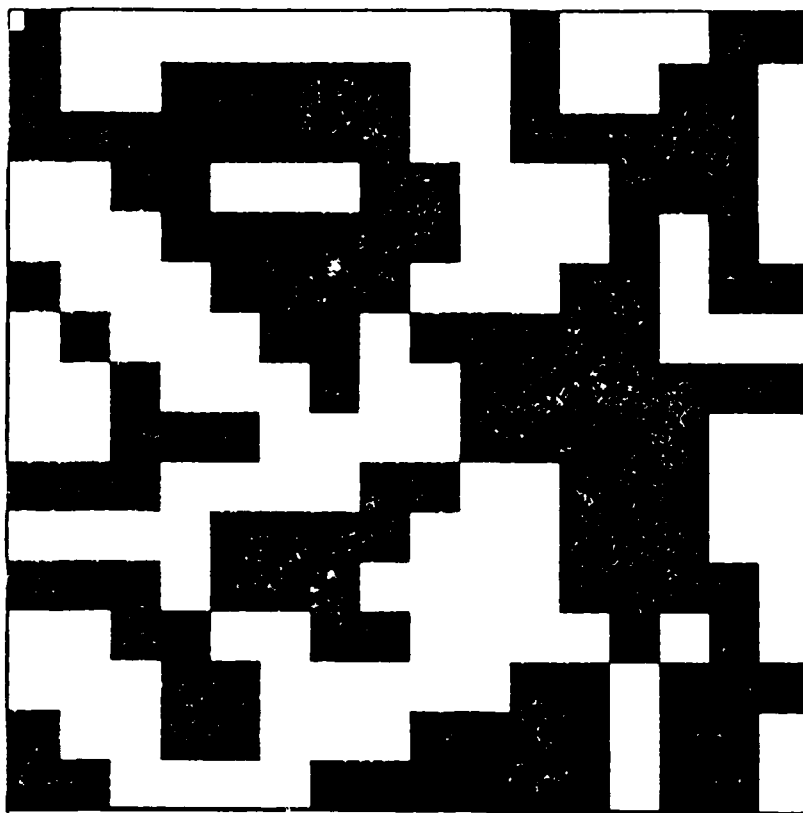
which relates the gradient  $(\Delta x, \Delta y)$  at point  $(x, y)$  to the  $r, \Theta$  parameters describing a line through that point and perpendicular to the gradient. This parameterization avoids problems with infinite slopes, and allows for any line segment in an image to be represented using parameters with finite ranges.



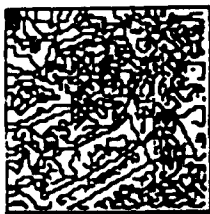
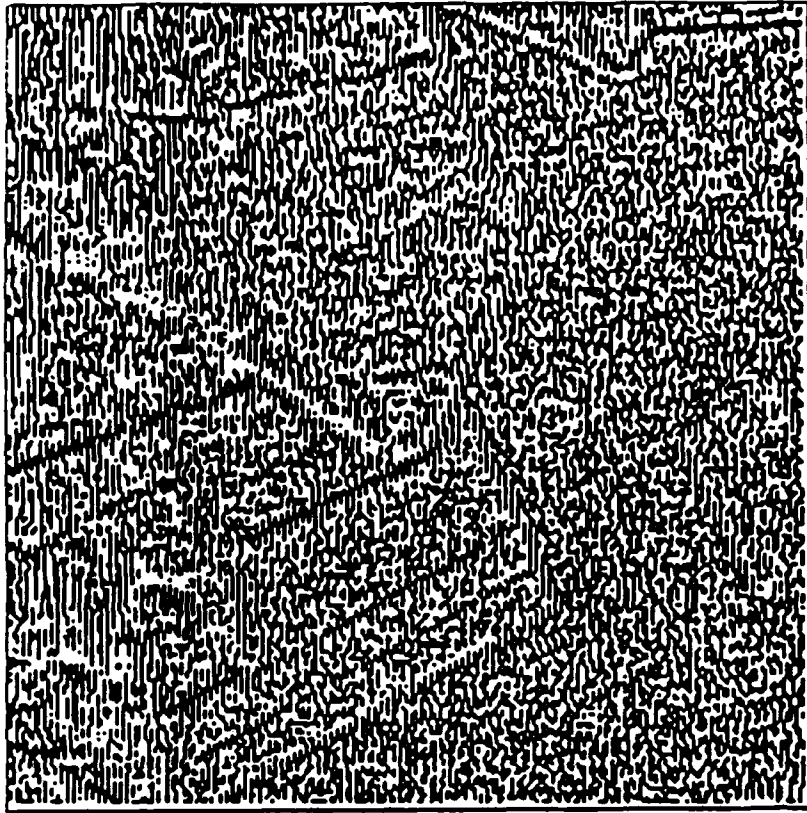
**Figure 4-16: Zero-Crossing Regions from the Laplacian Pyramid**



**Figure 4-17: Zero-Crossing Regions from the Laplacian Pyramid (cont'd)**



**Figure 4-18: Zero-Crossing Regions from the Laplacian Pyramid (cont'd)**



**Figure 4-19: Thresholded Contours**

Figure 4-21 shows the 1D projection of the Hough Transform for the image in Figure 3-2 from Section 3 onto the  $\Theta$  axis. It thus describes the distribution of  $\Theta$  values for gradient vectors from the image independent of  $r$ . The two peaks correspond to the general grid pattern in the image. Figure 4-22 shows the image points corresponding to one of the peaks in this histogram. The linear characteristics of the selected gradient values are apparent. Nonetheless, effective use of the Hough Transform requires further processing to group and thin these selected gradient points into distinct edges.

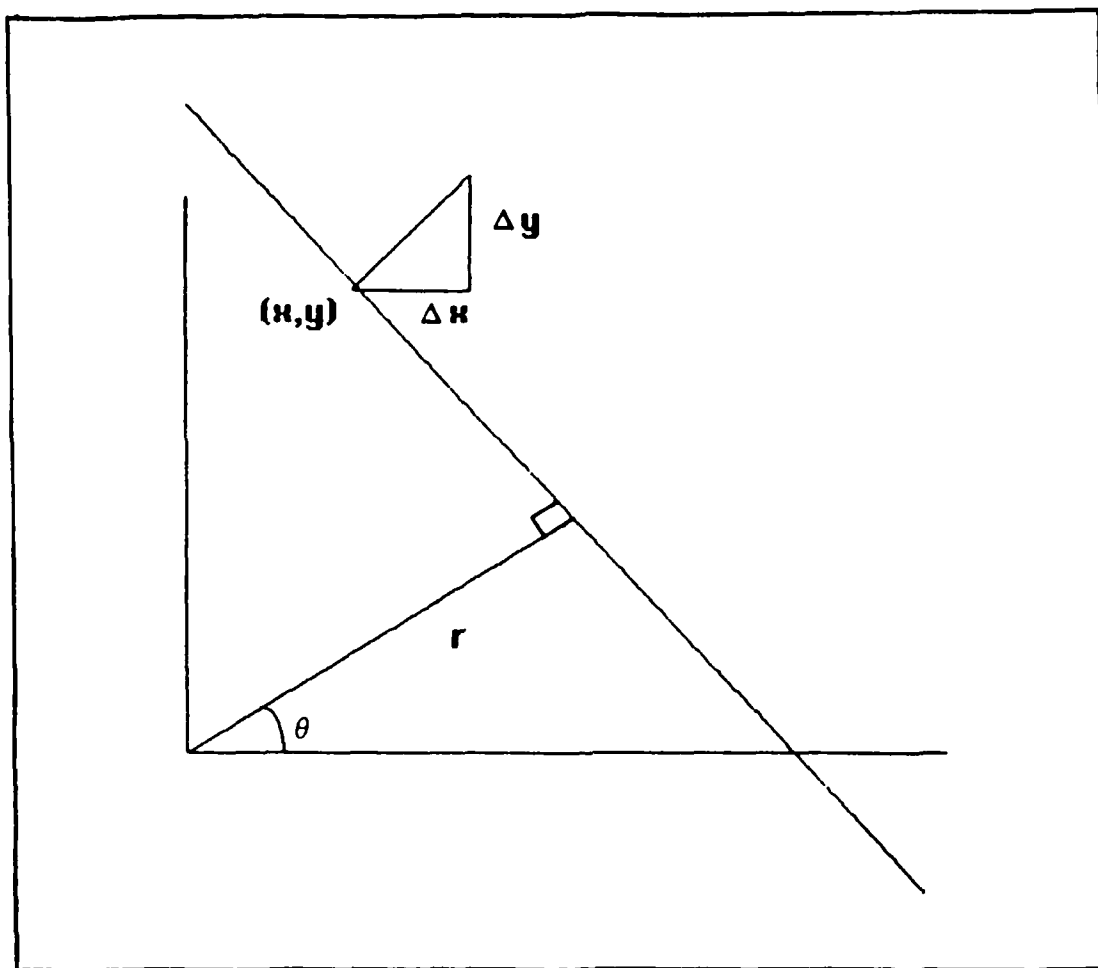
#### **4.2.5 Gradient Based Edge Linking**

Gradient Based Edge tracking techniques connect local measures of the image gradient together guided by criteria corresponding to such things as the length, shape, and contrast of the resulting connected edge. These procedures involve measuring the gradient at an image point to determine the local contour orientation. In this form of edge-tracking, image points correspond to nodes in a search tree, with arcs between nodes corresponding to a contour connection between image points. The expansion of the search tree is guided by general search techniques [Nilsson - 80] using evaluation measures based upon minimal or constant change in orientation or curvature.

We decided against gradient based edge tracking in favor of grouping over the segments and linear subsegments produced by the Canny, Burt, and region extraction segmentation routines. We found gradient based techniques were too local and not easily generalizable to linking based upon semantic criteria while the grouping process over extracted entities in the ISDB was. We did experiment with different ways of measuring the gradient support about a point by computing gradient deviation along different distances perpendicular to the gradient at a point (corresponding to contour length) and at distances along the gradient (contour width).

#### **4.2.6 Segment Based Edge Linking**

Segment based edge linking links together the edge segments generated by different edge operators using rather general conditions. For example, in tracking along a river, we want to link using two parallel edge segments, which surround a darkened area, are within some distance of each other, and have both local and



**Figure 4-20: Hough Transform Parameterization**

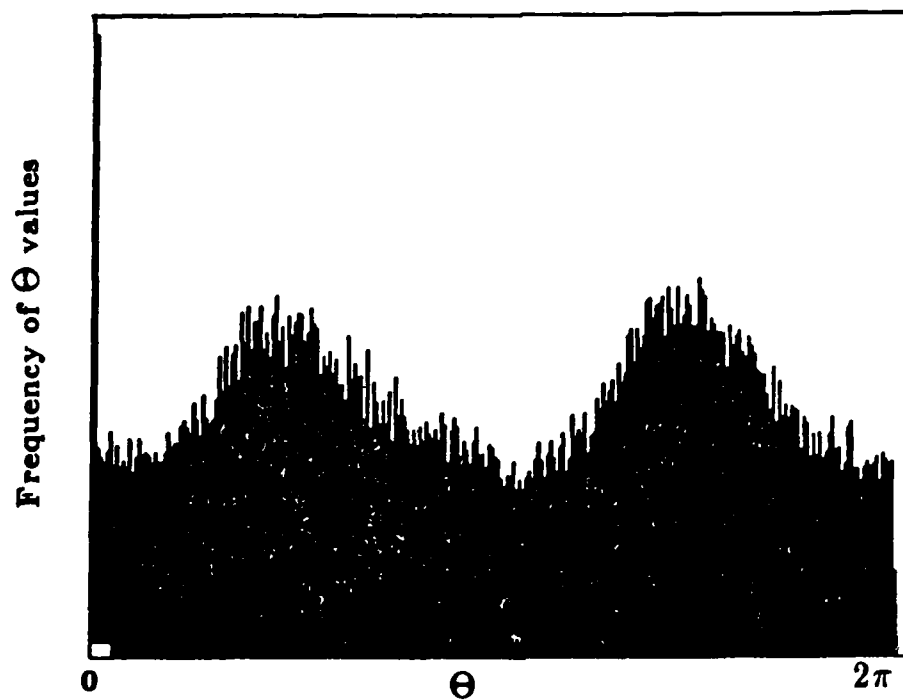


Figure 4-21: Orientation Histogram

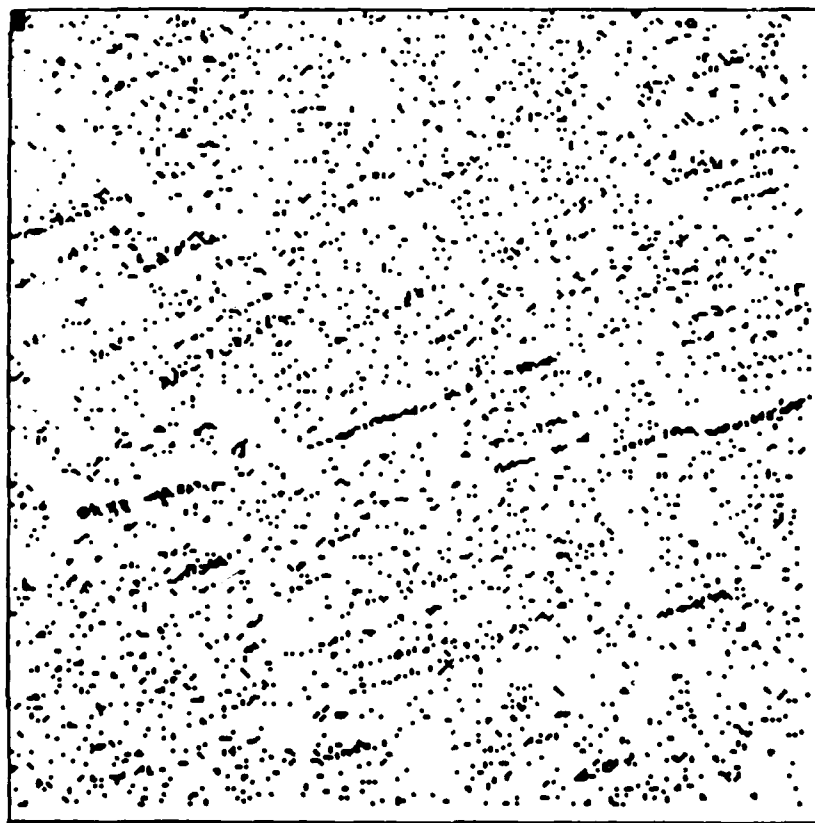


Figure 4-22: Selected Image Points Corresponding to Peak in Orientation Histogram

global constraints on orientation change with respect to contour length. Such processing involves operating over extracted structures and their associated relationships and attributes in the ISDB and Hypothesis/Task Data Base. An example is to extend from a selected curve segment and a direction, successive edge segments with minimal orientation change over a neighborhood. Figures 4-23 through 4-26 show this for a selected linear subsegment from a set of such segments. The key to this approach, in contrast to gradient based edge linking, is that the grouping can be made conditional on abstract relations and attributes, associated with entities in the ISDB, and related to the ongoing interpretation process.

Three things are involved in segment based edge linking:

- 1) A Successor Function which determines for a given edge and edge sequence what the allowable successors fragments are. Currently, our successor function generates all edge fragments contained in a given set of areas (Figure 4-27) parameterized by orientation and distance, using either *the current edge segment* or *the sequence of edge segments* along a given hypothesized contour.
- 2) An Evaluation Function which evaluates the correspondences of a set of edges to a connected curve sequence. The evaluation function is used to determine which curve from the successor set is best for extending a potentially connected curve sequence. There is an unlimited number of evaluation functions corresponding to different Programmed Finders and Segmentation Routines. One is to select the edge which is closest and is also within certain bounds of orientation, average contrast and intensity of the last selected curve segment. The linking can also be based upon more abstract geometrical characteristics of the curve, such as an approximation to a constant change in orientation; or a combination of these and that the edge is within some distance of a region with river-like attributes.
- 3) The Search Control which keeps track of the multiple curve sequences and determines which curve to continue linking processing upon. Starting from a given edge fragment, multiple sequences are possible. These

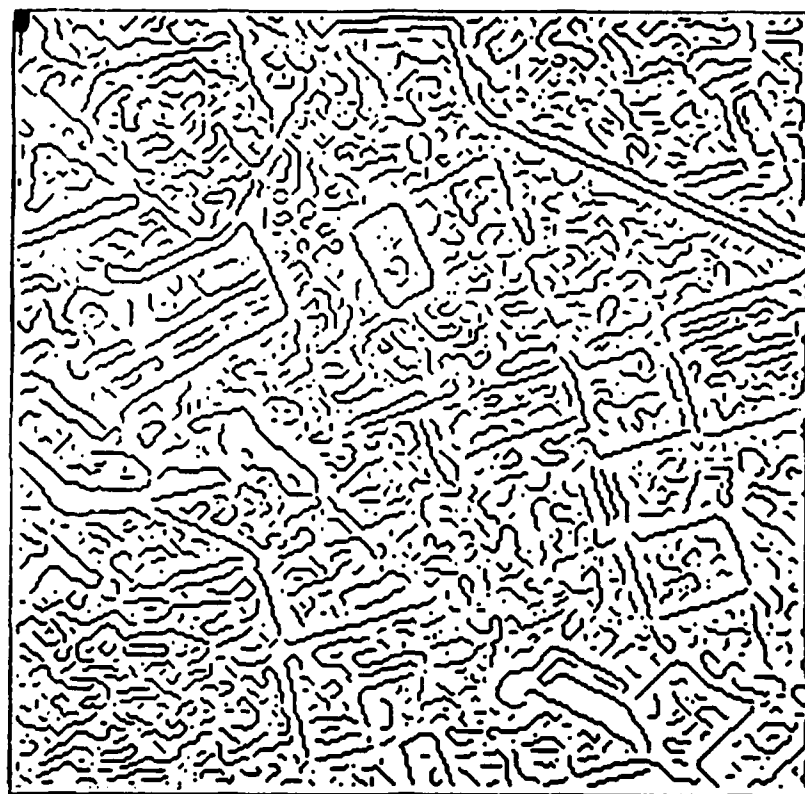
are maintained in a search tree organized by a global ordering from the evaluation function.

#### **4.2.7 Nevatia/Babu**

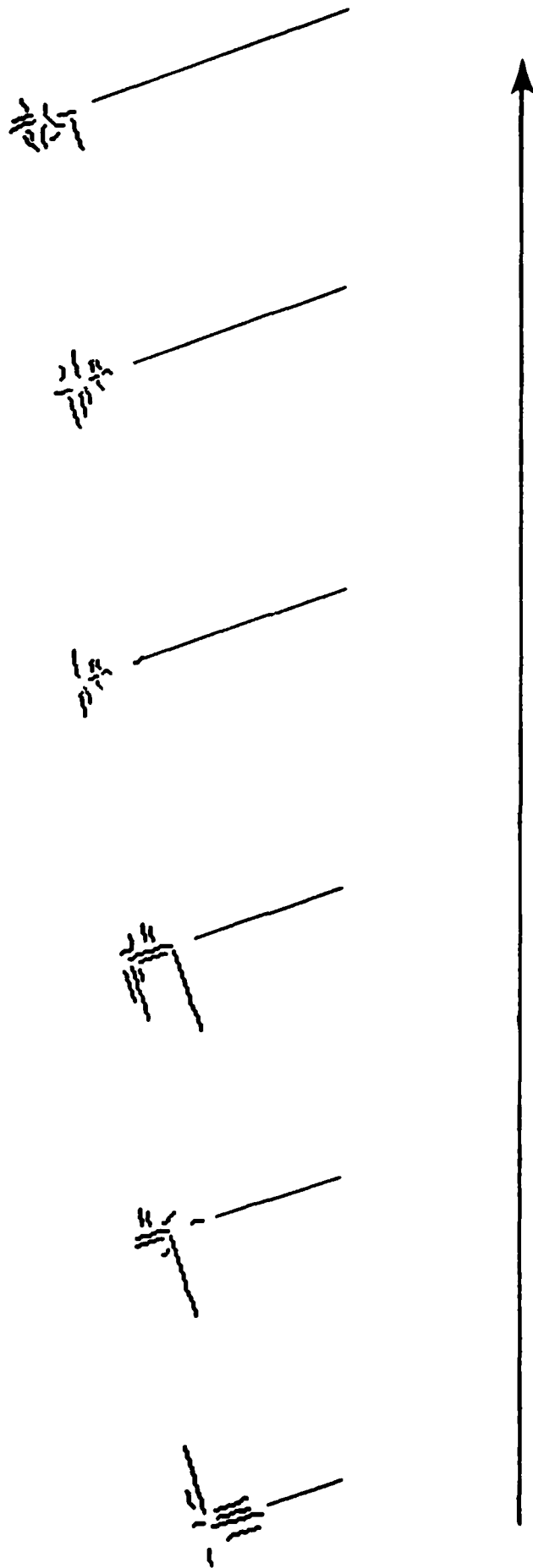
The Nevatia-Babu Line finder consists of the following steps:

- 1) Detect local edges by convolution with 5x5 masks sensitive to edges oriented in six directions (every 30 degrees) and store the mask convolution which gives the greatest response and the direction of this response.
- 2) Select these local edges based upon a threshold on their response magnitude and using a thinning procedure which selects an edge if its magnitude is greater than the neighboring pixel's edge magnitude in the direction perpendicular (non-maxima-suppression)
- 3) There is then a linking procedure operating over the selected edge points which extracts chains, forks, loops, isolated points, bridges. These are then fit to piece-wise linear segments to extract straight lines.

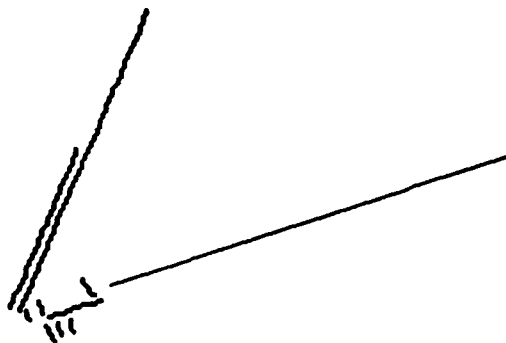
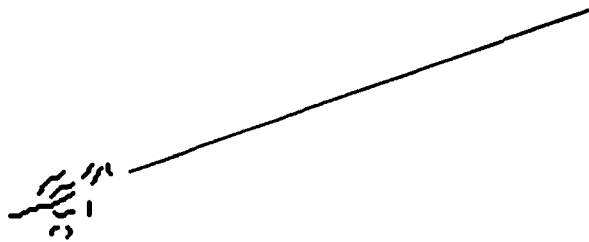
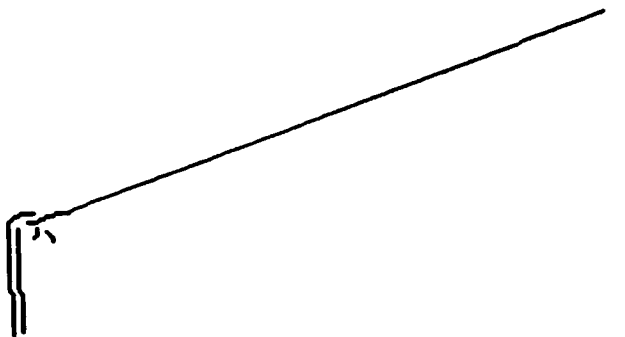
Our experience with the Nevatia Babu line finder is that it works extremely well at determining large features surrounded by sharp step edge of high contrast, but that the general characteristics of SAR (scintillation, side-lobing and scattering) degrade its performance. In general, there are an unyieldly number of parameters., especially for the third step. The applications of these operations should be dependent on the status of the ongoing interpretation process. Figure 4-28 shows a SAR image. Figures 4-29 and 4-30 show the outputs from the Nevatia-Babu edge finder using different thresholds.



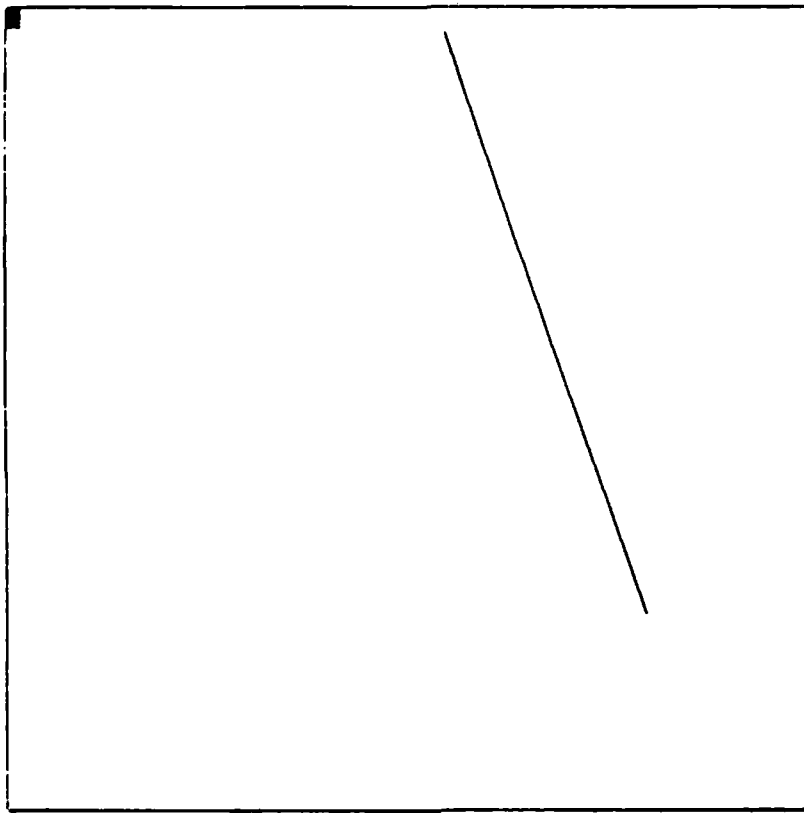
**Figure 4-23: Edge Segments**



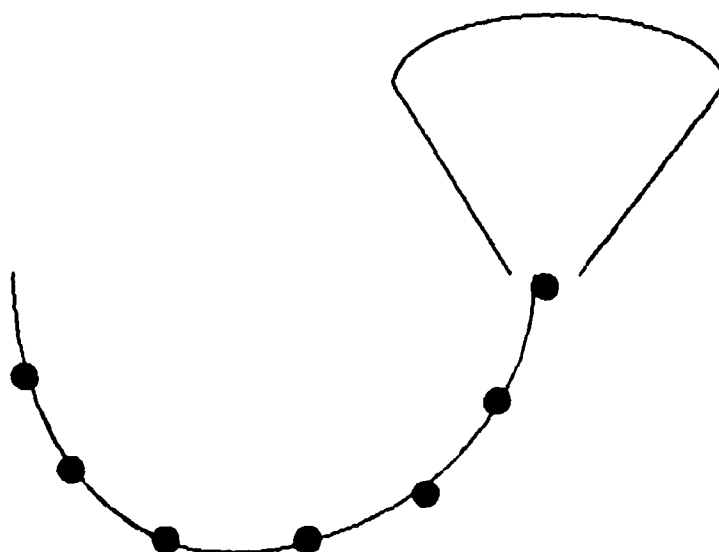
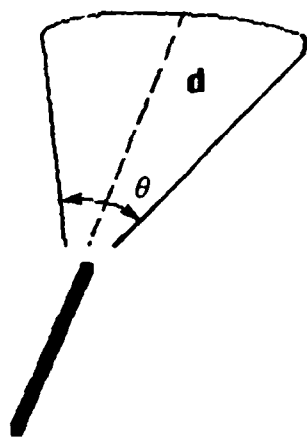
**Figure 4-24: Linked Edge Segments**



**Figure 4-25: Linked Edges Segments (cont'd)**



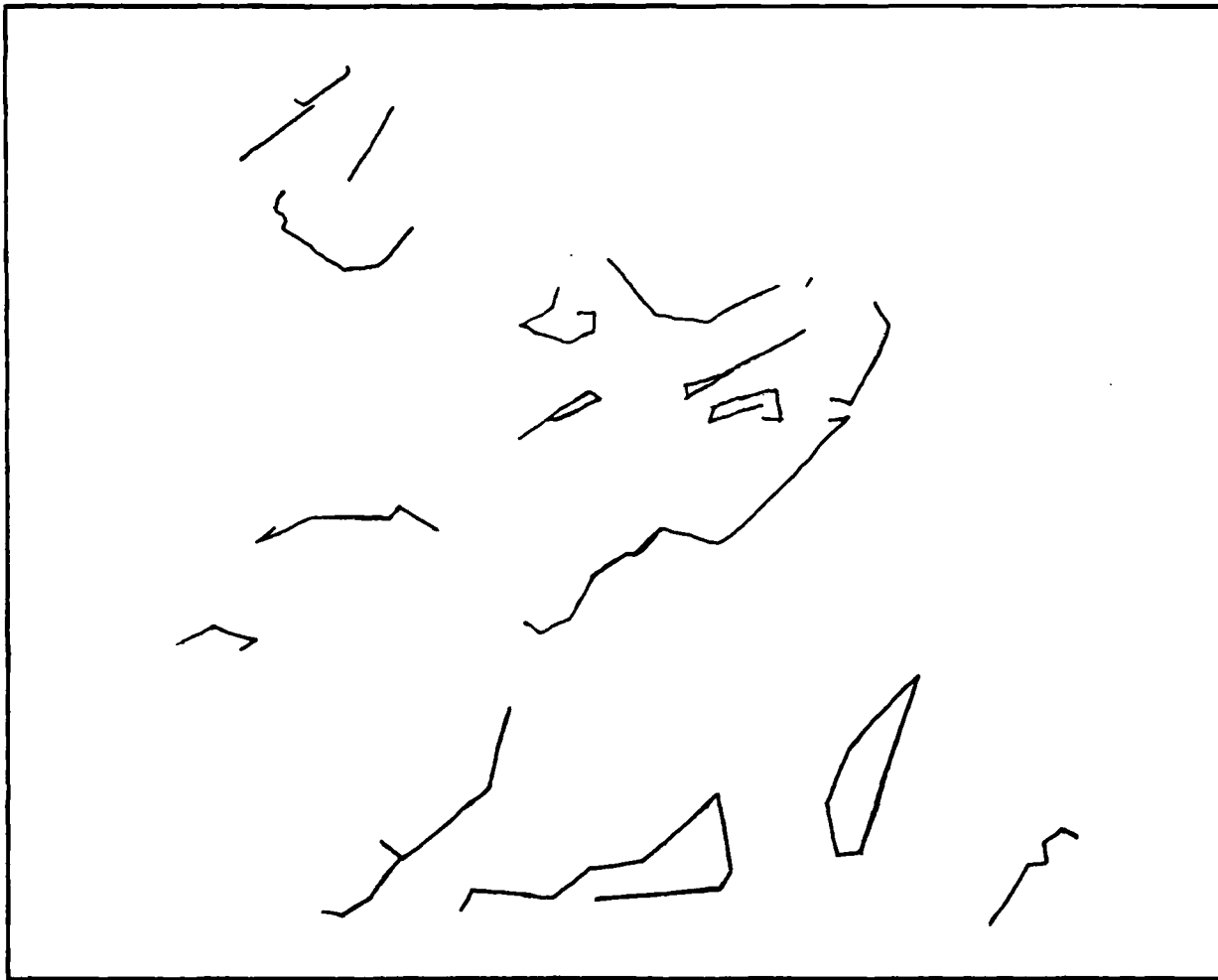
**Figure 4-26: Linked Edge with Respect to Extracted Segments**



**Figure 4-27: Successor Neighborhood**

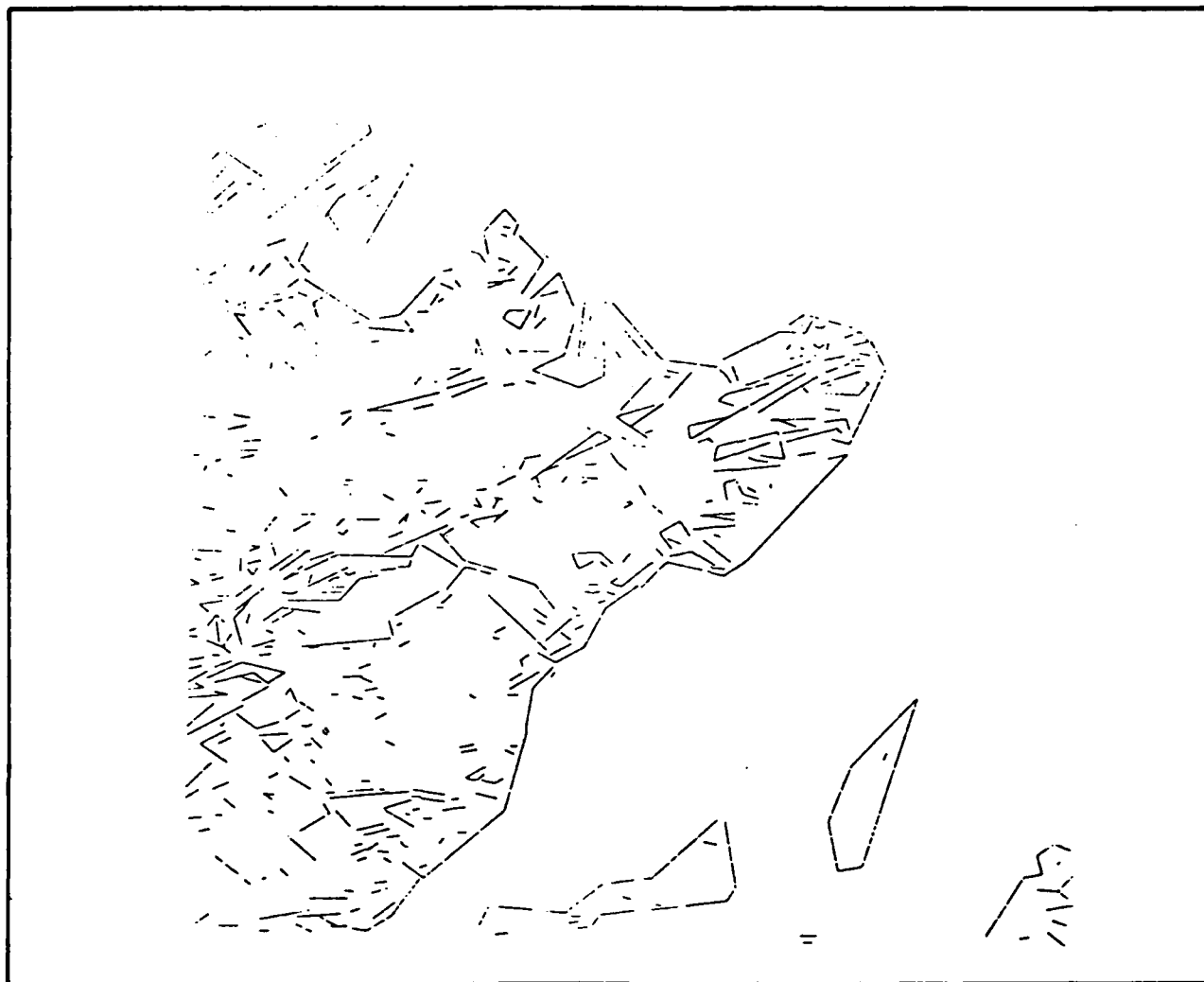


**Figure 4-28: Field Image: (ETL17)**



Edge magnitude threshold:	20000
Minimum number of pixels/chain:	20
Maximum epsilon of linear fit:	5

**Figure 4-29: Nevatia-Babu Edge Outputs**



Edge magnitude threshold: 2000  
Minimum number of pixels/chain: 5  
Maximum epsilon of linear fit: 5

**Figure 4-30: Nevatia-Babu Edge Outputs (cont'd)**

#### 4.2.8 Others

There were other edge extraction techniques which we studied but did not implement: Burns [Burns - 84] straight line fitting procedure and different relaxation based procedures [Hanson - 80]. In general, we have decided to use simple, understandable and controllable edge extraction processes, such as the Zero-Crossings, Burt and Canny to find edges as different spatial frequencies and strengths. More complicated processing will be performed by explicit grouping operations over these extracted structures stored in the ISDB. These grouping operations are based upon predicted object identity or general perceptual criteria expressed as segmentation rules and strategies.

### 4.3 REGION SEGMENTATION

Regions are connected image areas determined by the similarity of attributes reflecting texture or intensity or the gradient of such measures. As with edges, the region extraction processes should be tunable and the relations between parameters describing their operation and their effects explicitly understood. Also similar to edges, there are grouping and merging operations applied over extracted regions for joining or breaking them based upon shape properties, registration with an extracted linear feature, or the conditional evaluation of a weak difference in feature type between adjacent regions. These operations should be explicitly represented as hypothesis which can be evaluated and verified over time.

There are several region extraction processes which have such properties. Among them are conventional histogram-guided segmentation techniques [Ballard - 82] (and particular variants for relaxation updating of region labels [Nagin - 79] and application over image sub-areas [Kohler - 83]) and Burt's Hierarchical segmentation processing [Burt - 83a]. These processes may be applied over simple intensity and contrast measures or texture measures based upon such things as Markov coefficients, probability distributions, concurrency tables, and fractal dimension estimates.

#### **4.3.1 Histogram-Based Segmentation**

Histogram-based region segmentation is a general technique for breaking an image into connected areas with similar attributes. The basic steps are to obtain a histogram over an image or image area with respect to some set of features, extract clusters in the histogram, project these cluster labels back onto the image, and finally extract the connected label sets in the image. There are several variants of this basic procedure at each of these basic steps. In recursive segmentation techniques [Ohlander - 75], the extracted image areas become the image areas over which succeeding histograms are formed. In relaxation-based techniques [Nagin - 79], the histogram label image is modified by local pixel compatibilities. There are several different criteria by which clusters can be extracted from histograms. In fact, for higher dimensional feature histograms, the recognition of clusters can be as complicated as the recognition of structure in the underlying image itself.

We have used the simplest of these: 1D histograms over selected features in masked areas where peaks are extracted based upon being a local maxima over some range, separation from neighboring peaks by some distance, and the existence of a similarly distinctive minima point between peaks. We have also begun using some of the shape fitting procedures described in Section 4.4.1 to characterize peak structure in 1D histograms.

#### **4.3.2 Plurality Updating**

Plurality Updating involves changing the segmentation-attribute label associated with a pixel based upon the label values in a neighborhood surrounding the pixel. In Plurality Updating, as the name implies, this involves going with the majority label in the neighborhood. The effects of Plurality Updating are to smooth out segmentations locally.

#### **4.3.3 Texture Segmentation**

The Hierarchical Discrete Correlation, or HDC, is similar to the Burtian Pyramid except the resolution stays constant from level to level and the elements of the 5x5 convolution mask elements are applied to pixels separated by greater and greater distances. The result is a rapid technique for computing image

properties, centered at a point over larger and larger neighborhoods. The HDC is useful for texture classification, since, for a given attribute such as edge orientation, contrast, or density it can compute the average value directly and variance by subtracting HDC values from different levels.

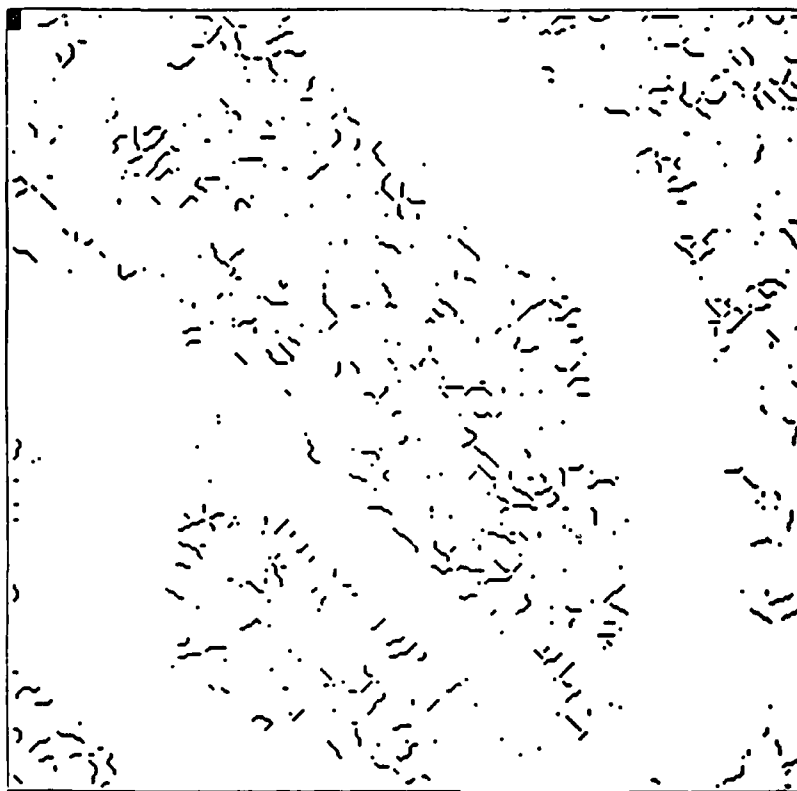
Figure 4-31 shows selected edge elements based upon contrast and length. Figures 4-32 through 4-34 show contour plots of the HDC at higher and higher levels based upon average edge density.

Related to the HDC, and which we have found to be of more use, is decomposing an image into same size sub-images and computing attributes over the image structures in the ISDB which are contained in the image sub-areas. This yields an object-based, multi-resolution description of image properties, which is more controllable than the HDC. An example of this technique was given in Section 2 where segmentation was done on edge fragments of a certain length with respect to intensity. The size of the sub-images can be directly related to the type of environmental feature being used for texture classification.

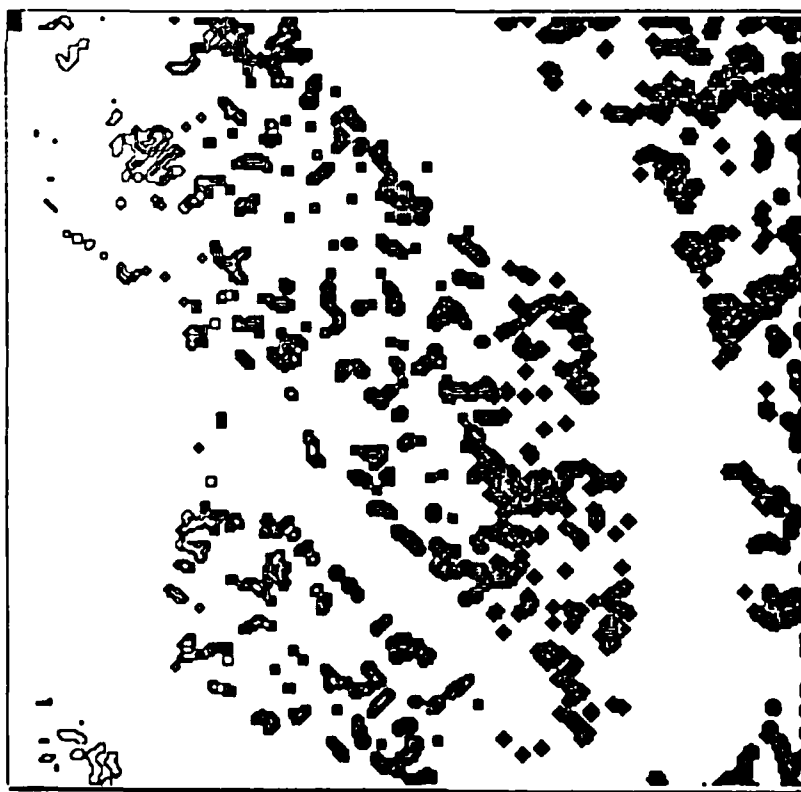
This type of texture segmentation, especially with SAR imagery, can also occur with respect to regions extracted by thresholding. Figure 4-35 shows The effects of sensor resolution on region texture elements extracted by thresholding. The texture classification occurs using the region attributes of the blobs over image sub-areas.

#### **4.3.4 Kohler**

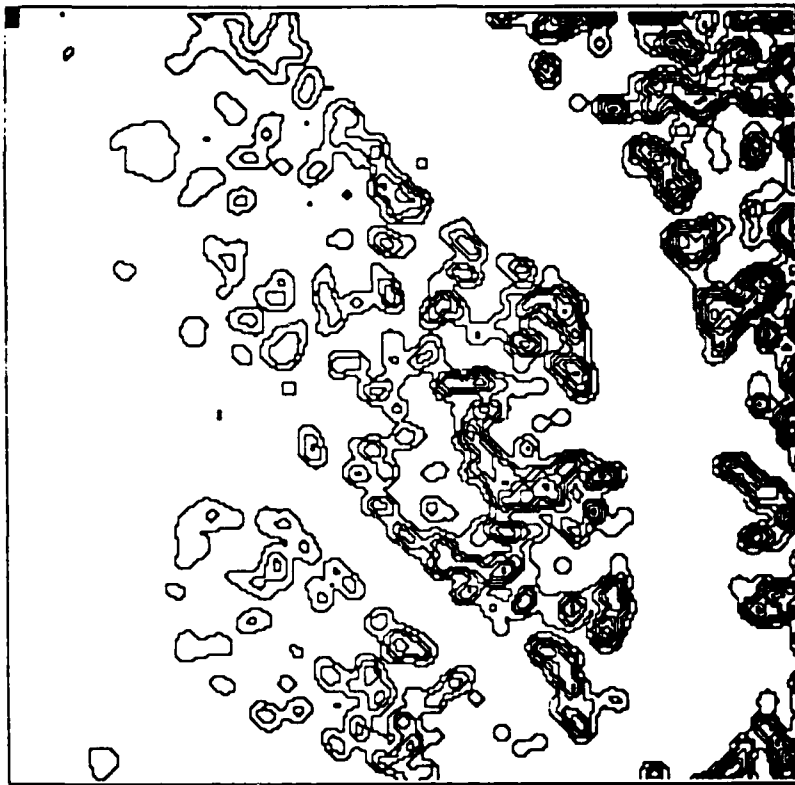
Kohler's Segmentation Procedure [Kohler - 81] is a histogram-based procedure which takes into account local edge structure and contrast. An edge-element between image pixels will vote for a value in the histogram if a threshold at that value places an edge between the associated pixels. This vote can be modified by the relative value of the pixels and the threshold. Thus, a threshold can be selected which maximizes/minimizes contrast, or the number of edges, or ratios of these two. Peak detection in Kohler's algorithm is simplified since the maximal value in the histogram is always selected. Edges selected by this value are removed from further consideration when the procedure is repeated.



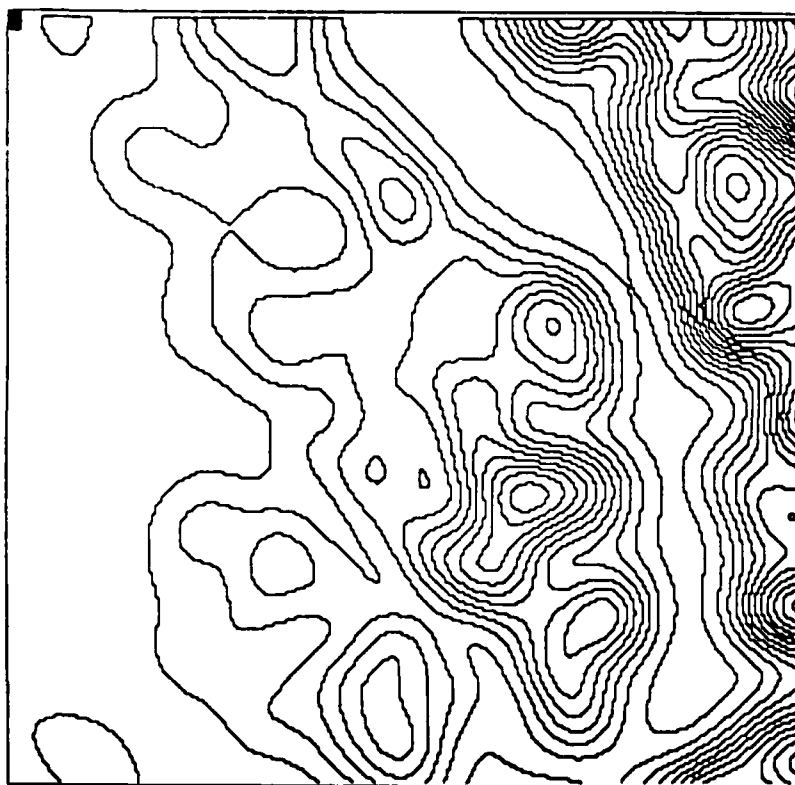
**Figure 4-31: Selected Edge Fragments**



**Figure 4-32: HDC - Level 1**

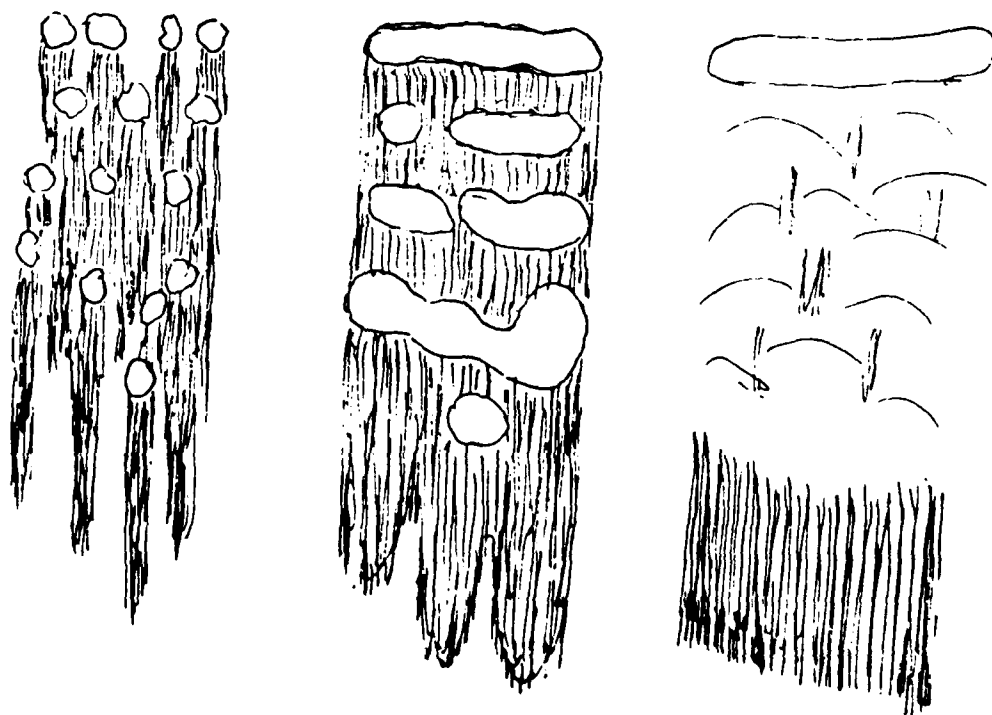


**Figure 4-33: HDC - Level 2**



**Figure 4-34: HDC - Level 3**

Resolution (ft)	< 10	≈ 10	> 10 (20-30)
Feature			
Tree line at leading edge of forest	Individual non-shadowed trees	Bright line oriented with forest edge	Bright line oriented with forest edge
Trees within forest	Tree shape plus shadow with measurable spacing	Bright pixel(s) plus shadow with estimatable spacing	Highly convolved making Fourier methods most useful
Shadow at trailing edge	Tree outlines	Outlines of tall trees	Dark line with width proportional to average tree height



**Figure 4-35: Region Based Forest Texture Invariants**

## 4.4 SHAPE EXTRACTION PROCEDURES

### 4.4.1 Extraction of Significant Curvature Points

An essential task is decomposing an extracted edge into a sequence of sub-contours based upon some shape fitting requirement, as in approximating a curve by linear segments, polynomials, or splines. This involves extracting points of significant orientation change or curvature. An example of a contour and different linear approximations is shown in Figures 4-38 and 4-40.

Recursive line fitting is based upon evaluating the linear fit of a line relative to a given curve using a point along the curve with maximum distance from the line segment. Such a point is used to generate two new line segments which are each evaluated with respect to their points of maximal distance. This procedure is repeated, recursively, to each generated linear approximation until the linear approximations are all within some distance of their associated curves. For a closed contour, the initial points are selected to be immediately adjacent. One of these points is discarded when the final fit is achieved. Figure 4-36 shows the set of extracted edges from ETL-Image-17 (Figure 4-28 in Section 4.2.7). Figure 4-37 shows a selected set of these edges based upon average contrast. Figure 4-38 shows a selected edge from this set and the recursive line fits to it.

The other method is based upon approximating curvature at points by a local, iterative, procedure we developed for implementation in a parallel array architecture [Lawton - 85]. The technique begins by associating an *orientation value* with each point along a contour. The orientation value may come directly from the image gradient. The orientation value at each point is then updated by averaging it with those of the immediately adjacent points. The number of iterations of this averaging process corresponds to weighted evaluation of curvature over different neighborhood sizes along a contour. *Interesting* points are then extracted where significant changes and variations in orientation occur as reflected by the *neighborhood difference measure*. In its undirected scalar form, this measure is the sum of the absolute differences between the orientation value at a point and its immediate neighbors. The interesting points are the local maxima in this measure which also exceed some threshold. Figure 4-39a-d shows the positions of these points for a given connected contour with different amounts of

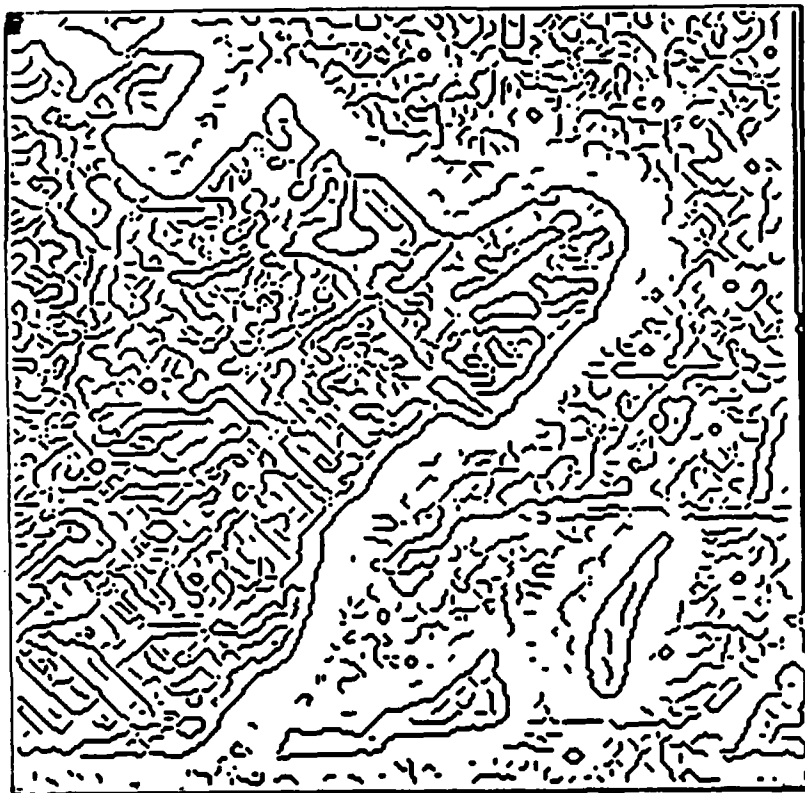
averaging. Figure 4-40a-d shows the corresponding linear interpolation. Figure 4-41a-d shows the interpolated, smoothed contour from the resulting orientation values. Figure 4-42a-d shows the orientation values along these contours for different amounts of smoothing.

The shape of a contour is also described by the changes in orientation in the sequence of linear segments along it. The histogram of orientation values along the contour is useful as a statistical description of shape.

#### **4.4.2 Chamfer-Based Shape Descriptions**

Image Chamfering [Barrow - 78] is used in a variety of basic matching and shape characterization applications. Chamfer processing associates with each point in an image, an approximation to its minimal Euclidean distance from a boundary. Figure 4-44 shows a set of boundaries extracted from the image in Figure 4-43. Figure 4-45 shows the contour lines of the chamfer generated from this. As one moves from the river segment boundaries, the chamfer values increase. Thus, the chamfer image could be used to determine the average distance of some object from the river. The chamfer image can be used to compute the attributes of image structures: the average and variance of chamfer values along a curve characterizes its distance and orientation to another image structure. Chamfer generation is a two pass operation and requires local operations over 3x3 neighborhoods similar to median filtering. Chamfering is used for matching extracted edge structures for registering images using extracted contours or predicted contours from a model. It is also a basic source of information for generating shape descriptions.

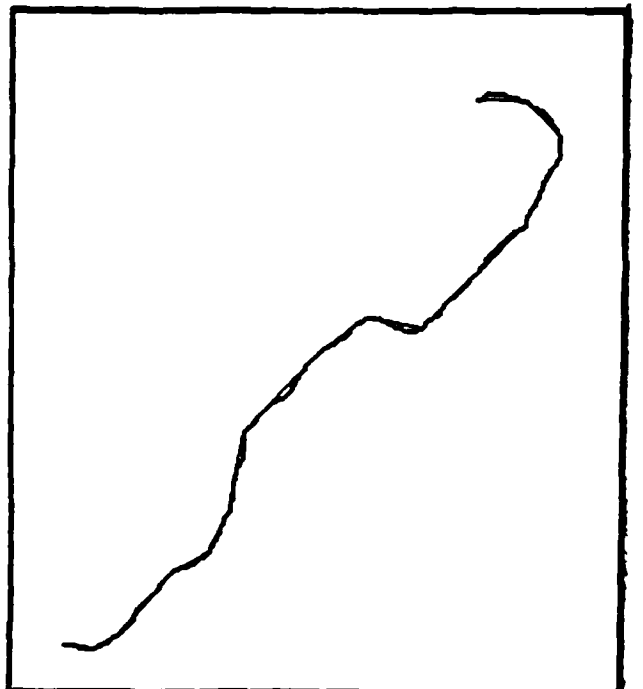
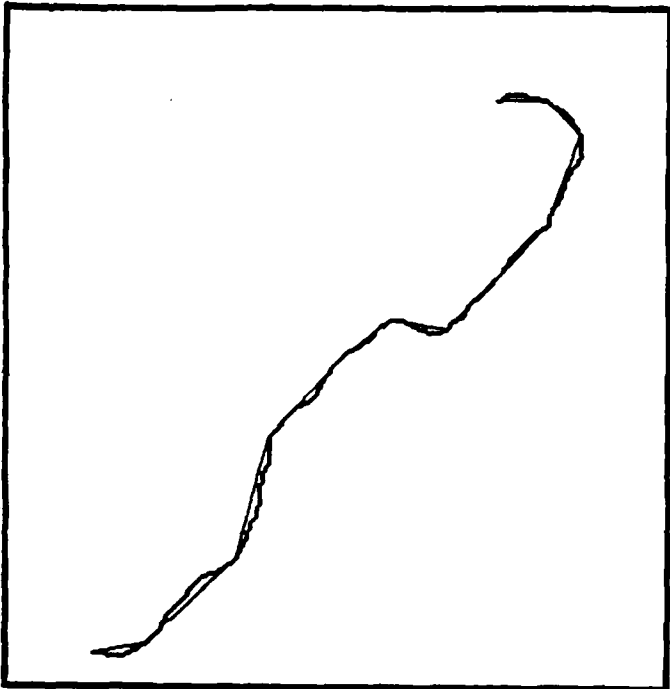
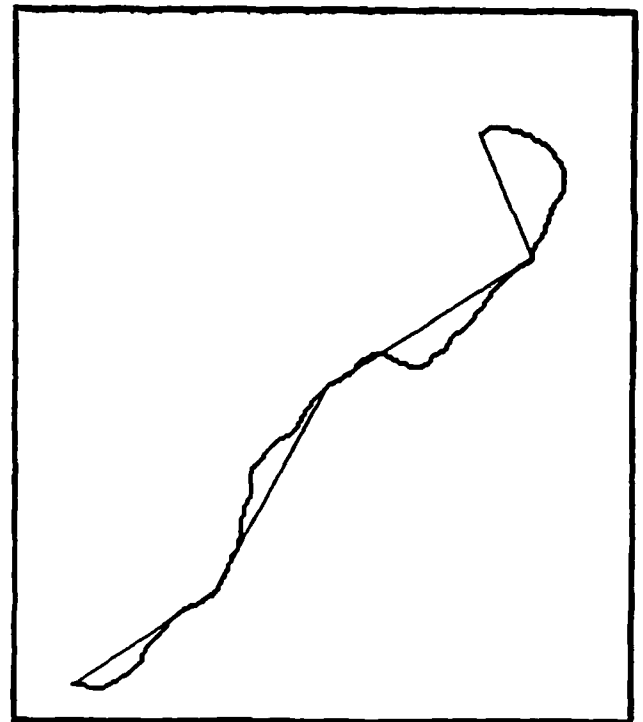
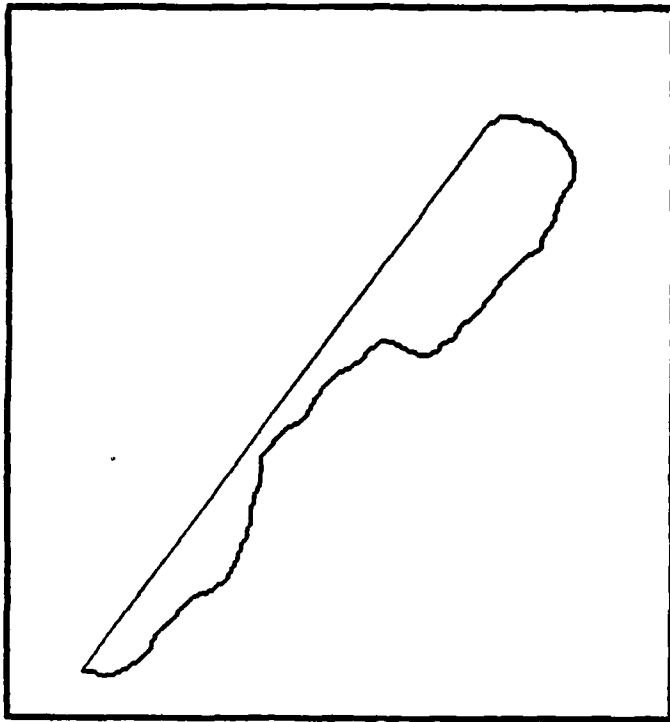
We have extended chamfer generation so that it associates not only the distance of an image point to the nearest image structure, but also the label of the nearest image structure. We refer to this as the Chamfer-Label Image and it is related to techniques which take the Laplacian of a chamfered image to determine the medial axis transform. In Figure 4-46 we see two labeled regions, A and B, and an image point P1. The chamfering labeling process associates the distance of a point to the nearest structure boundary, and also the label of that structure, as shown in the figure. Boundaries in the chamfer label image divide an image into regions where each is associated with the image structure to which any point in the region is closest. This is a discrete analog of the Voronoi



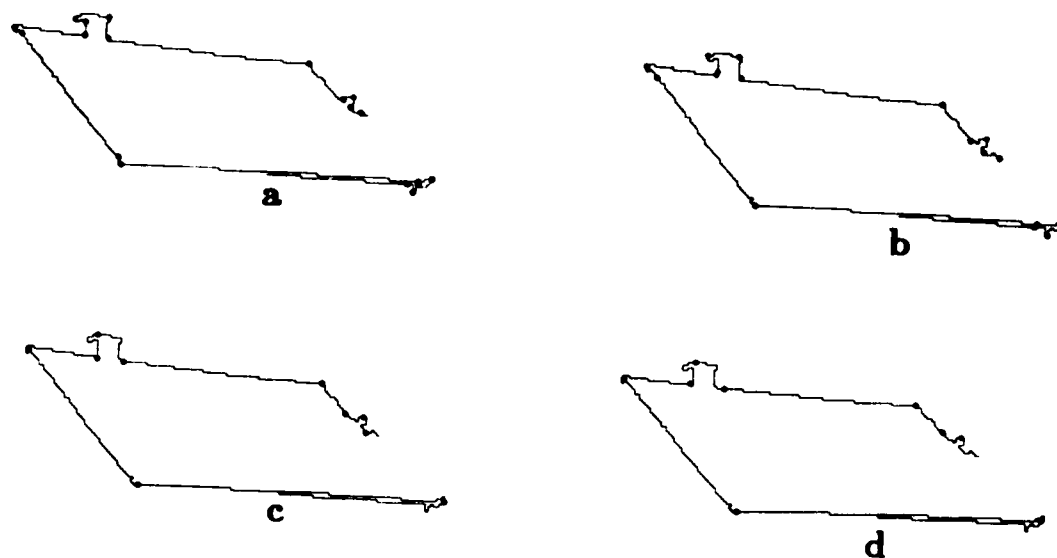
**Figure 4-36: Extracted Edges**



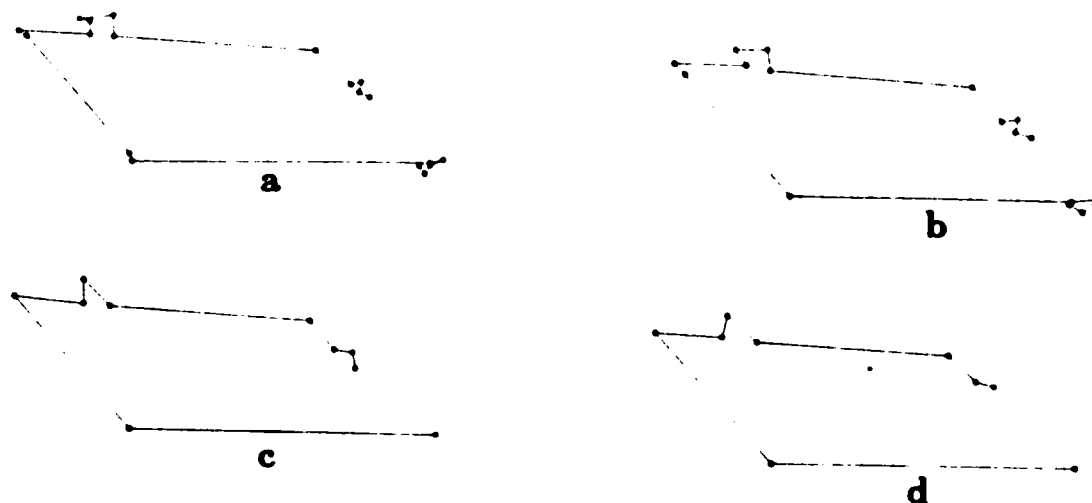
**Figure 4-37: Edges Selected on Average Contrast**



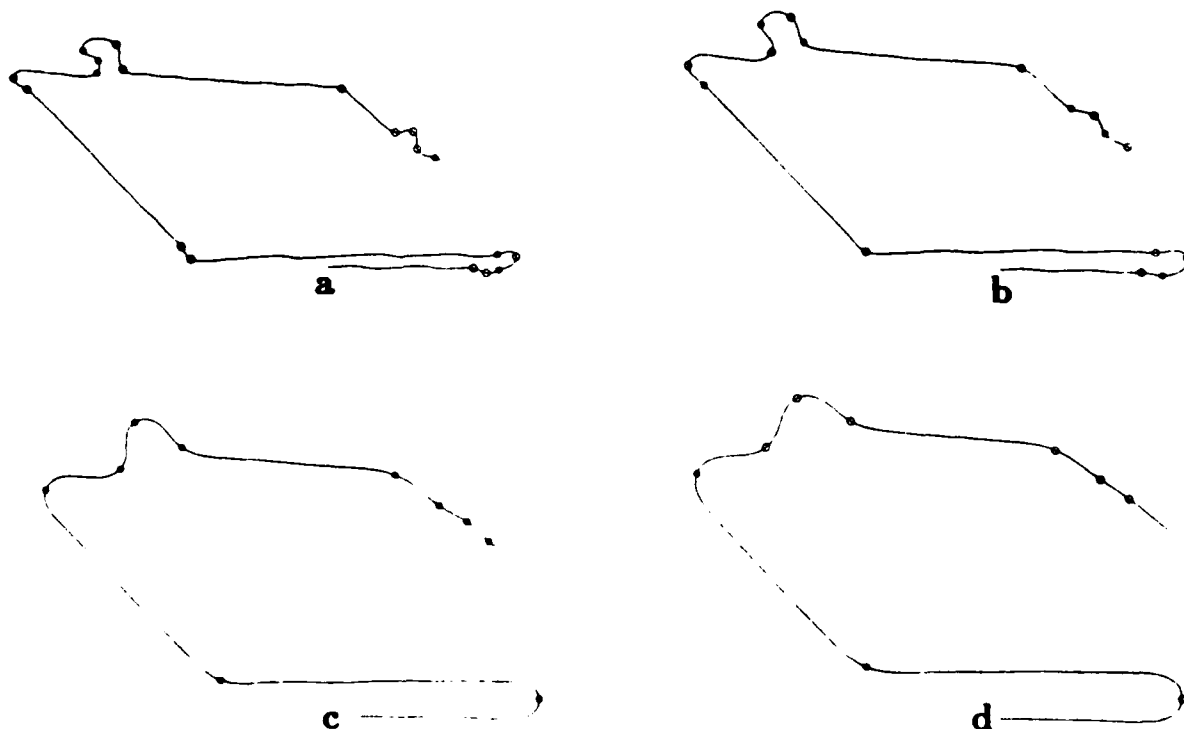
**Figure 4-38: Recursive Line Fit Linear Approximations**



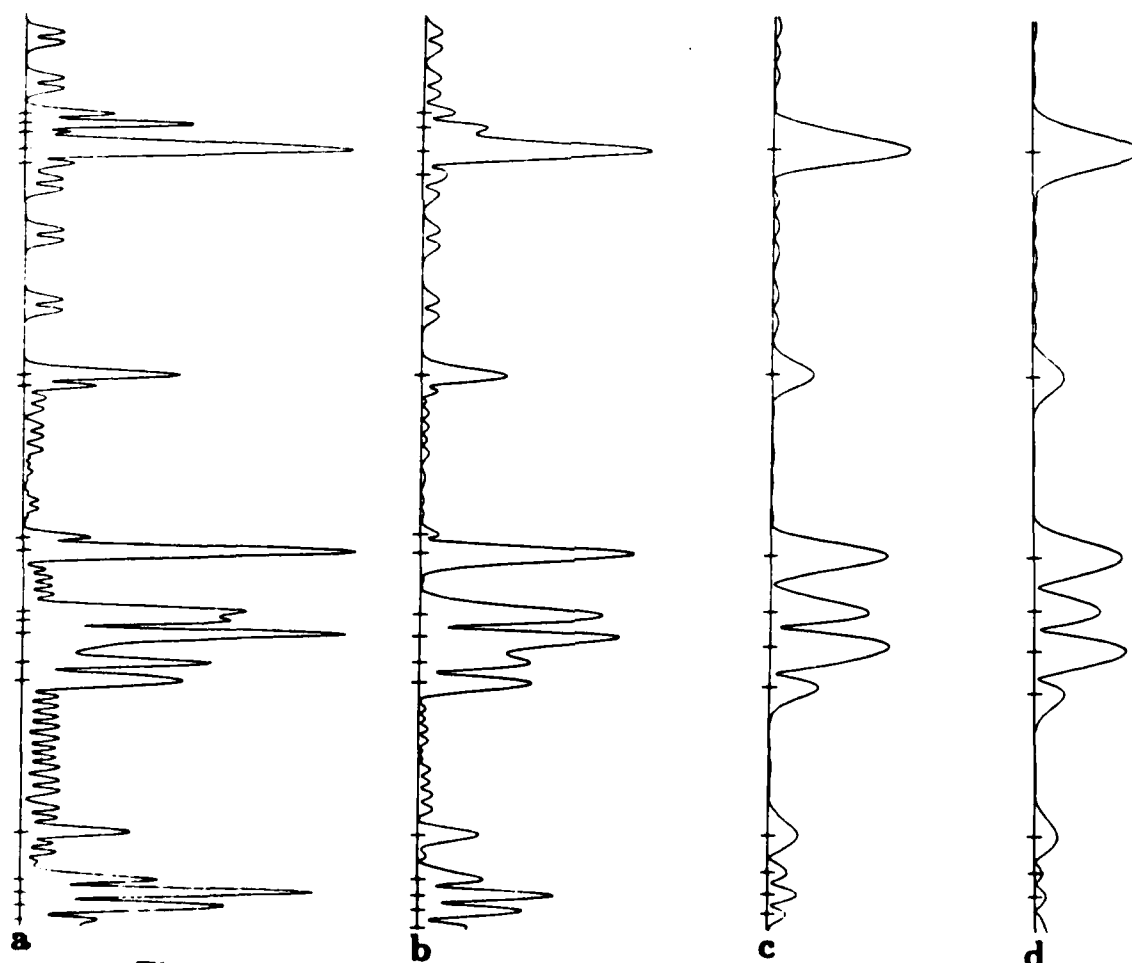
**Figure 4-39: Interesting Points Positioned  
with Respect to the Original Contour**



**Figure 4-40: Linear Segments**



**Figure 4-41: Interpolated Smoothed Contour**



**Figure 4-42: Orientation Differences Along Contour**

Diagram. Figure 4-47 shows a set of extracted boundaries from a river region which are labeled. Figure 4-47 also shows the boundaries between the chamfer labels generated from these two boundaries. Note how the elongated parallel boundaries are indicated by a chamfer label boundary between them. That these boundaries are parallel is indicated by the low variance in the chamfer values along the chamfer label boundary between them.

This use of chamfer labeling generates an exoskeleton describing the relations between extracted regions or edges. It can be used to generate an inner-skeleton for a single region by breaking the region's boundary into subsegments and associating a unique label with each subsegment. This yields a structure similar to the medial axis transform [Blum - 67]. This skeleton associated with a region is a rich source of shape descriptions, as in finding major axis and offshoots and their orientations. It is a multi-resolution shape description when the region subsegments are formed using the techniques described for extracting significant curvature points at different resolutions or distance tolerances with the recursive line fitting procedure. Figure 4-48 shows the shape skeletons corresponding to the extracted significant contour points at different resolutions.

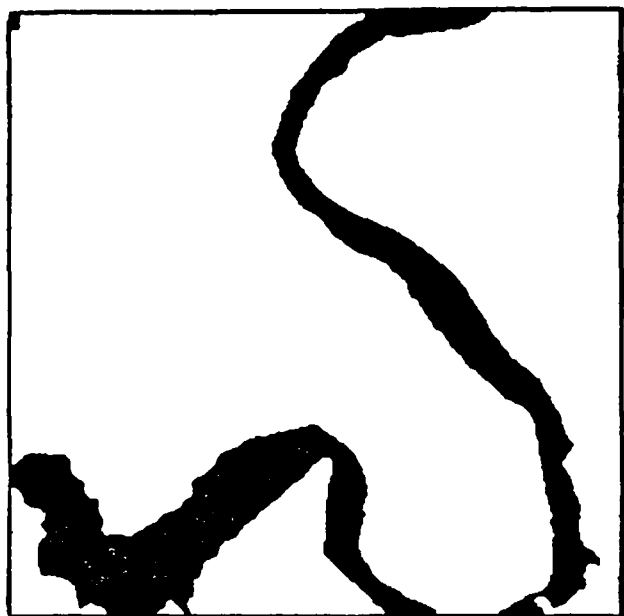
#### **4.4.3 Basic Shape Statistics**

There are several basic shape measures which are associated with regions. Among these are:

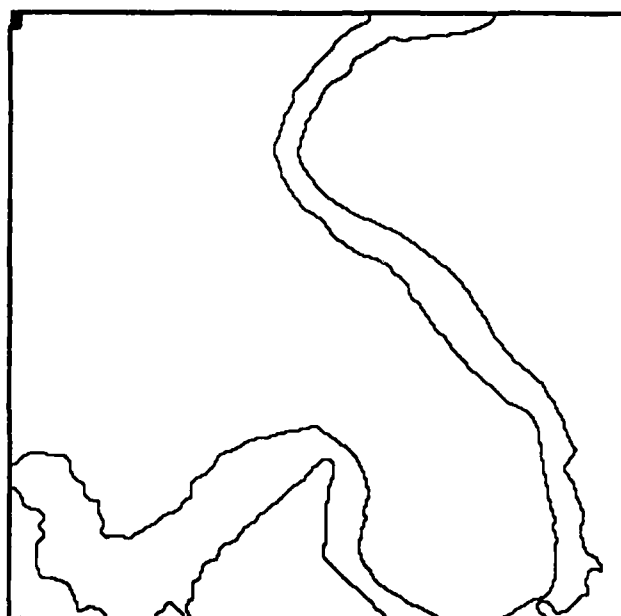
- Centroid
- Bounding Rectangle
- Moments
- Area
- Perimeter
- Topology (number of holes)

#### **4.5 SEGMENTATION RULES**

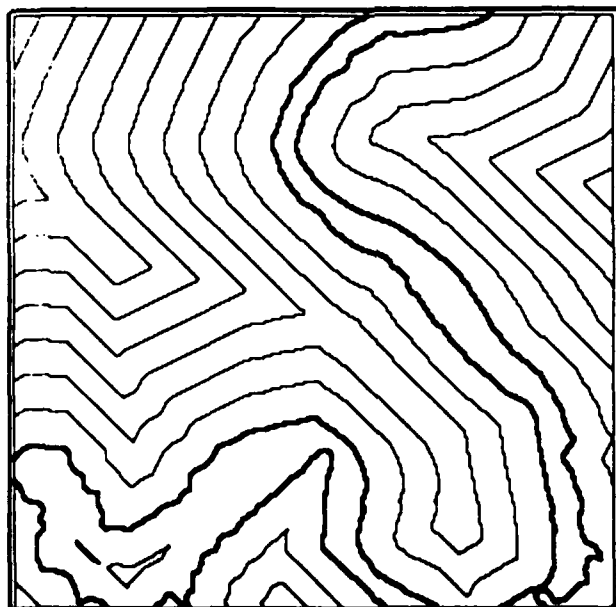
The application of the segmentation procedures is directed and implemented by a set of segmentation rules. These rules are organized to be run in a data-driven or model-directed fashion. These come in three general forms. *Extraction rules* which specify a sequence of actions to extract a particular type



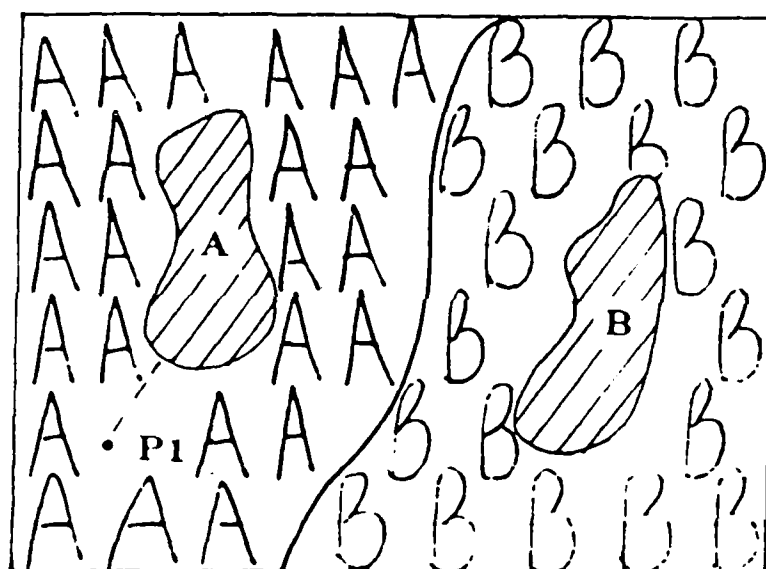
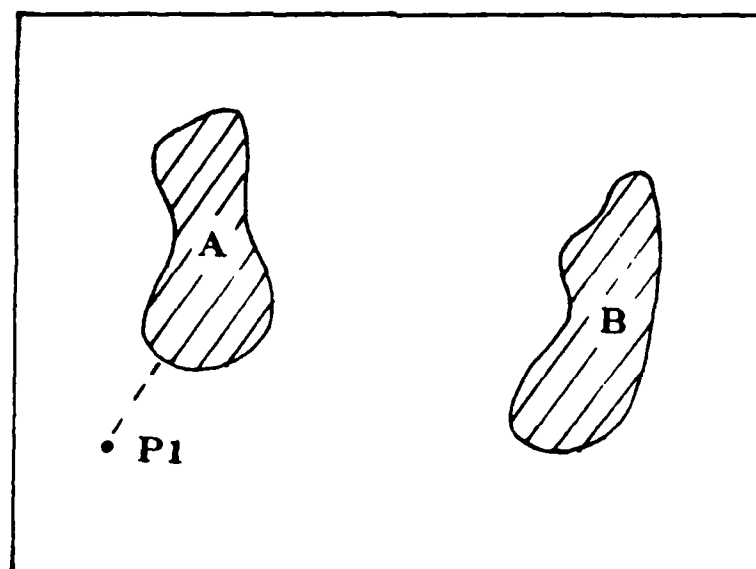
**Figure 4-43: River Segment**



**Figure 4-44: Extracted Boundaries**



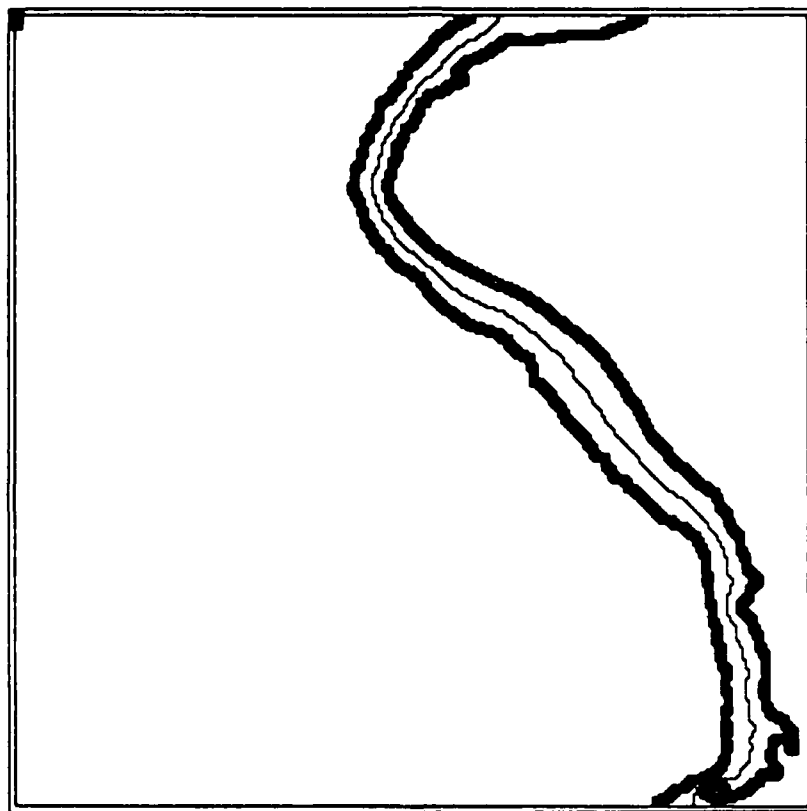
**Figure 4-45: Contour of Image Chamfer**



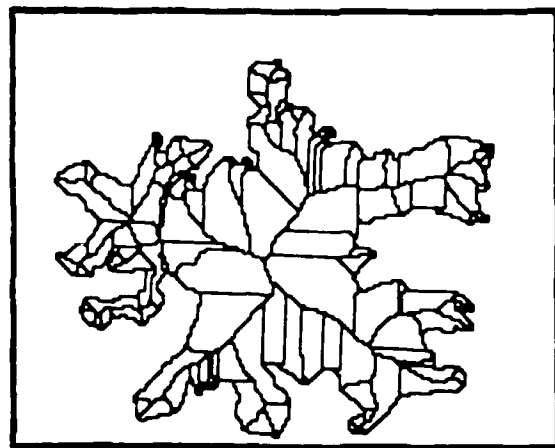
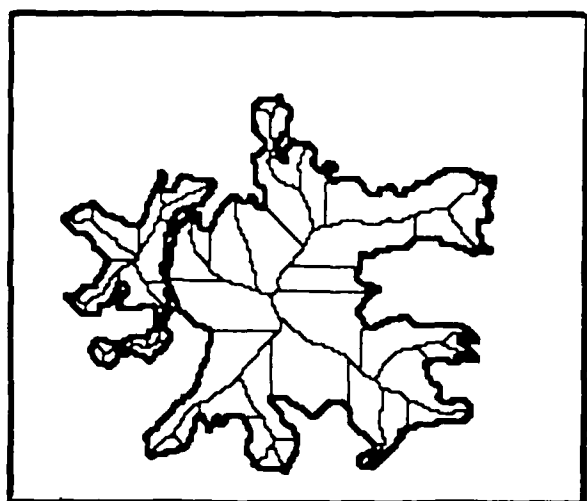
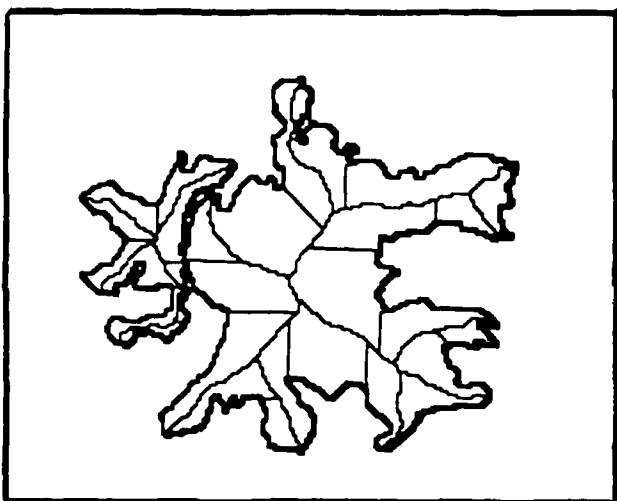
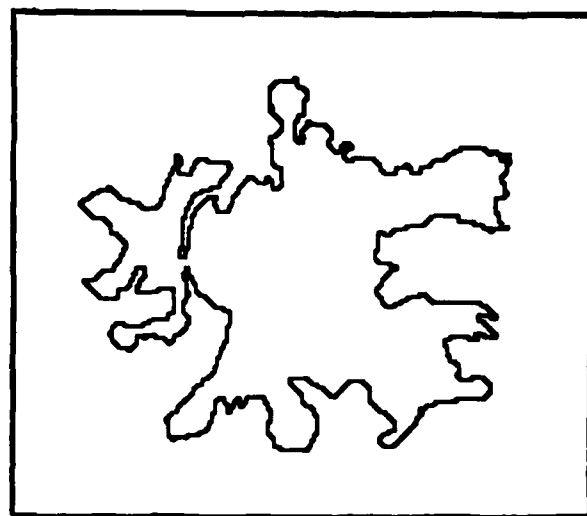
**FOR POINT P1**

- 1. DISTANCE TO NEAREST OBJECT (CHAMFER VALUE)**
- 2. THE LABEL OF THE NEAREST OBJECT (CHAMFER-LABEL VALUE)**

**Figure 4-46: Label Distance Chamfer**



**Figure 4-47: Chamfer-Label Boundary from River Segments**



**Figure 4-48: Multi-Resolution Shape Skeletons from Chamfer-Labeling**

of feature or image structure; *recognition rules* which determine when a particular structure or relationship exists; and *grouping rules* which generate a hypothesized structure from some relation between selected image structures. These basic rule types will be combined during application to produce a result with some specific quality. Thus, a river tracking procedure could involve calling an extraction type rule to pull out a particular type of image structure and then parameterizing a grouping rule to be sensitive to this type of structure.

A segmentation rule specifies:

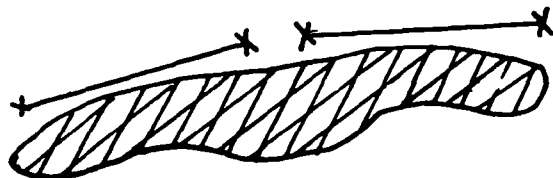
- The binding of rule variables to extracted image structures and hypothesis.
- A sequence of operations to perform and the associated binding of rule variables to image structures and hypothesis generated during rule application.
- A rule evaluation function to evaluate the success of the rule.
- Hypothesis/Tasks generated as a result of the rule and how to initialize their attributes (average shape of grouped regions).

Example segmentation rules are shown in Figure 4-49.

In general we found that simple region and edge extraction processes which could be applied in a focused and flexible manner were best. Processing is then built out of sequences of these operations expressed as rules for extracting particular types of image structure.

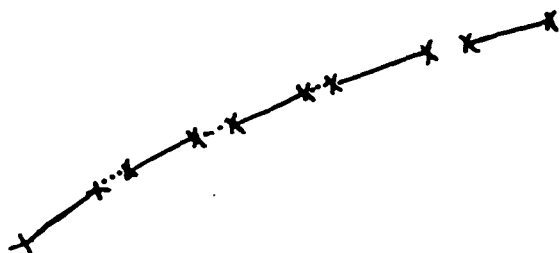
### Local Edge Linking

orientation difference  
average contrast  
similar region adjacencies



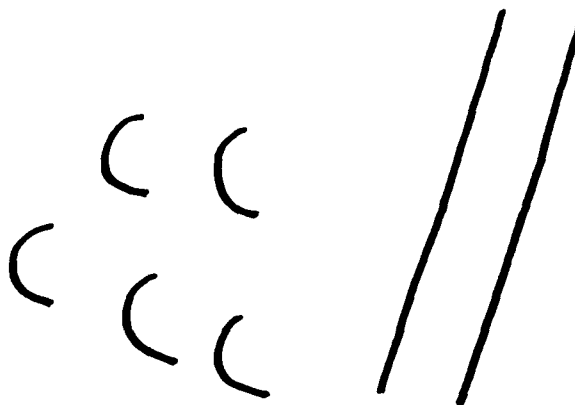
### Global Edge Linking (tracking)

properties along grouped  
contour



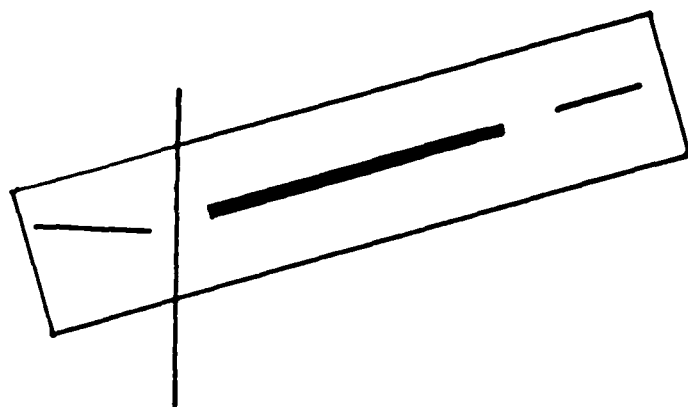
### Token Grouping

- parallel contours
- texture
- resolution filtering



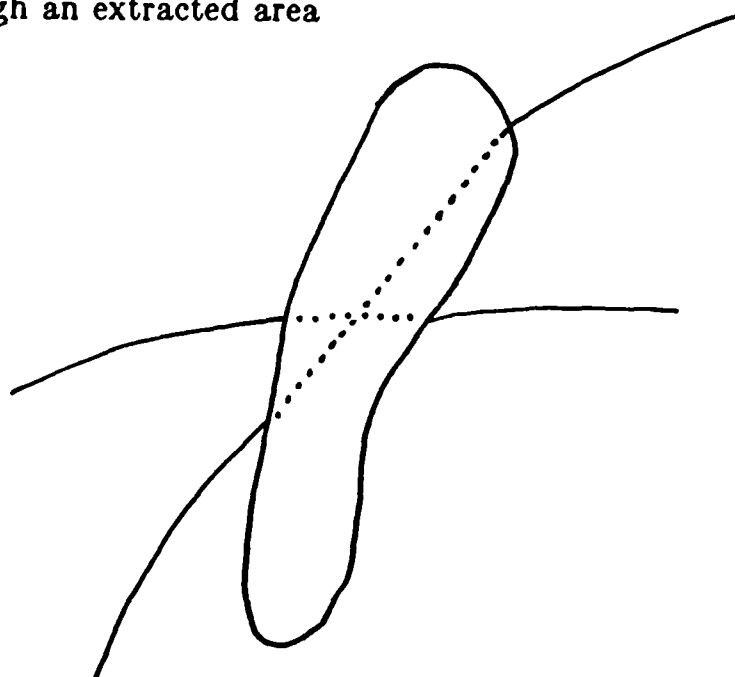
**Figure 4-49: Example Segmentation Rules**

### Global Linear Focusing Rules



- look for similarly oriented contours in this area
- look for intersections with other areas associated with other lines

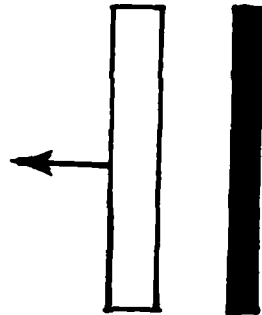
### Continuity through an extracted area



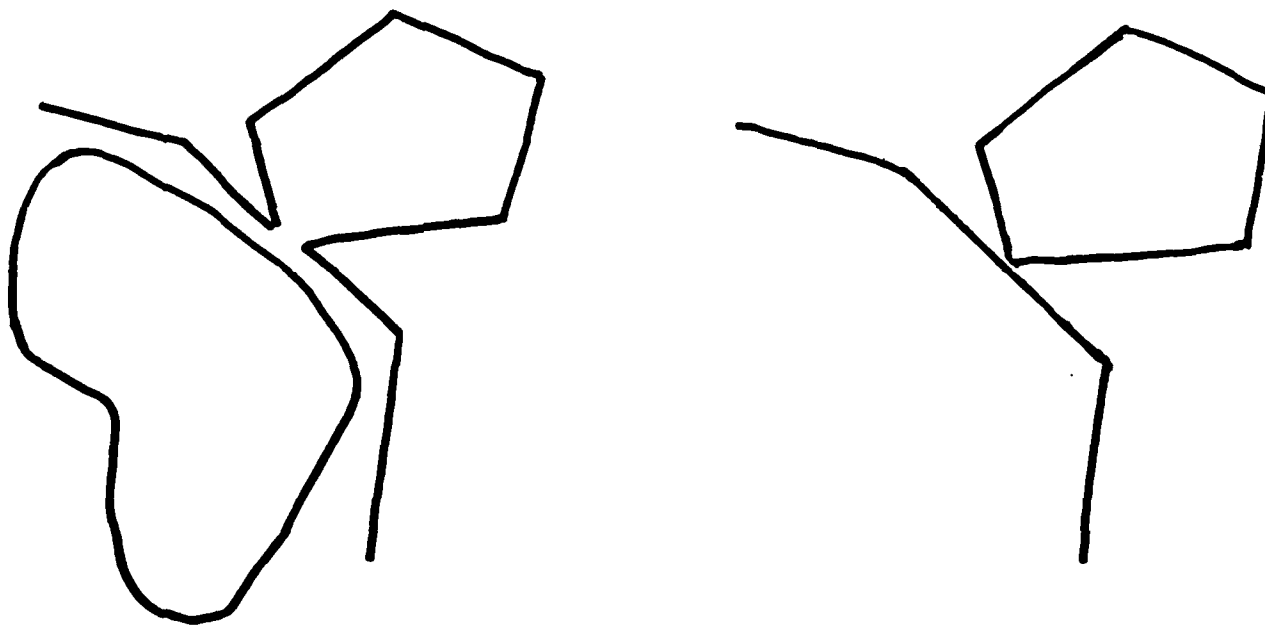
**Figure 4-49: Example Segmentation Rules (cont'd)**

### Strong Contour Non Suppression

- Search for weaker contours at higher frequencies in direction perpendicular to a strong edge



### Contour Busting



**Figure 4-49: Example Segmentation Rules (cont'd)**

## 5. SAR OBJECT KNOWLEDGE REPRESENTATION

The SAR Object Representation specifies the expected attributes and components of objects and the relationships between objects. It is used to associate object hypothesis with extracted image structures; to generate predictions for instantiating other related objects, and for specifying the validation process for instantiated hypothesis. The representation developed can be thought of as a network consisting of a set of nodes with each node corresponding to a particular object or object types, such as TERRAIN-AREA, FOREST-AREA, or LARGE-RIVER-SEGMENT. Each node contains two different types of descriptions: a declarative one describing the associated object in terms of its image properties and a procedural description, called a FINDER. The FINDER specifies how to extract such an object from an image in terms of particular segmentation routines. Objects are related by four general types of links which specify relationships and the inheritance/modification of attributes: IS-A, SIMILARITY/DIFFERENCE, COMPATIBILITY, and PART-OF relations [Tsostos-80, Tsostos-84]. Links are similar to nodes in that they have procedural and declarative attachments for describing and extracting the specified relationship.

The declarative object descriptions associated with nodes are used to match objects to image structures, especially during the initial instantiation of hypotheses. Once an object is instantiated as a hypothesis, the various links associated with it in the representation network are used to direct the instantiation of related objects and the ascription of certainty to the particular object instantiation. The certainty of an object hypothesis is reflected by the number of anticipated relationships for which there is evidence as determined by the links associated with the object which are themselves instantiated. For example, when an object is instantiated, its similarity/difference links are used to determine other potential objects which could correspond to the same set of structures but must be different in some specified way. Multiple objects can be instantiated with respect to the same sets of image structures.

The general structure of the declarative part of the object nodes consist of a set of specified queries over the ISDB and the HTDB and the expected results

or range of results to these queries. Often, as in implementing a feature vector approach, these queries will correspond directly to simple attributes of extracted image structures. Thus, a BIG-RIVER-SEGMENT is described as a large dark region, elongated with dominant parallel structure, and low contrast. These properties correspond directly to region attributes in the ISDB. The declarative object descriptions will also often correspond to the type of interesting structures which the segmentation knowledge source is trying to produce. The FINDERS specify for a given object, the types of segmentation procedures which are necessary for extracting it. These direct the segmentation processes to extract the related image structures.

Objects are related by four different types of links: IS-A, SIMILARITY-DIFFERENCE, COMPATIBILITY, and PART-OF. Each of these links contains declarative attributes and procedural attachments. The links are also instantiated during the interpretation process as instances of relations between objects in the HTDB. The properties of these links are:

**IS-A:** specifies the classification of objects and the structured inheritance of properties. The IS-A Links relating different types of terrain and water bodies is shown in Figures 5-1 and 5-2.

**SIMILARITY/DIFFERENCE:** specifies that two objects are alike with respect to some set of attributes or relations but different with respect to others. This isolates critical distinguishing features such as a river being like a road, except for different inherited network properties, different average curvature, and so forth. This link specifies the set of attributes, how they should differ, and particular programs to perform the disambiguation. Figure 5-1 shows the similarity/difference links between the FORESTED-TERRAIN area and the URBAN-TERRAIN area. Declaratively, this link contains information about the different types of texture and contrast between the terrain types. In general, sets of objects having the same parent through IS-A Links will be interconnected by Similarity/Difference Links to specify their distinguishing characteristics. This consists of the specific attributes which must be found to distinguish the objects or, procedurally, it can consist of procedures to be executed to evaluate the potential conflict between the objects.

**COMPATIBILITY:** specifies allowable and expected relations between

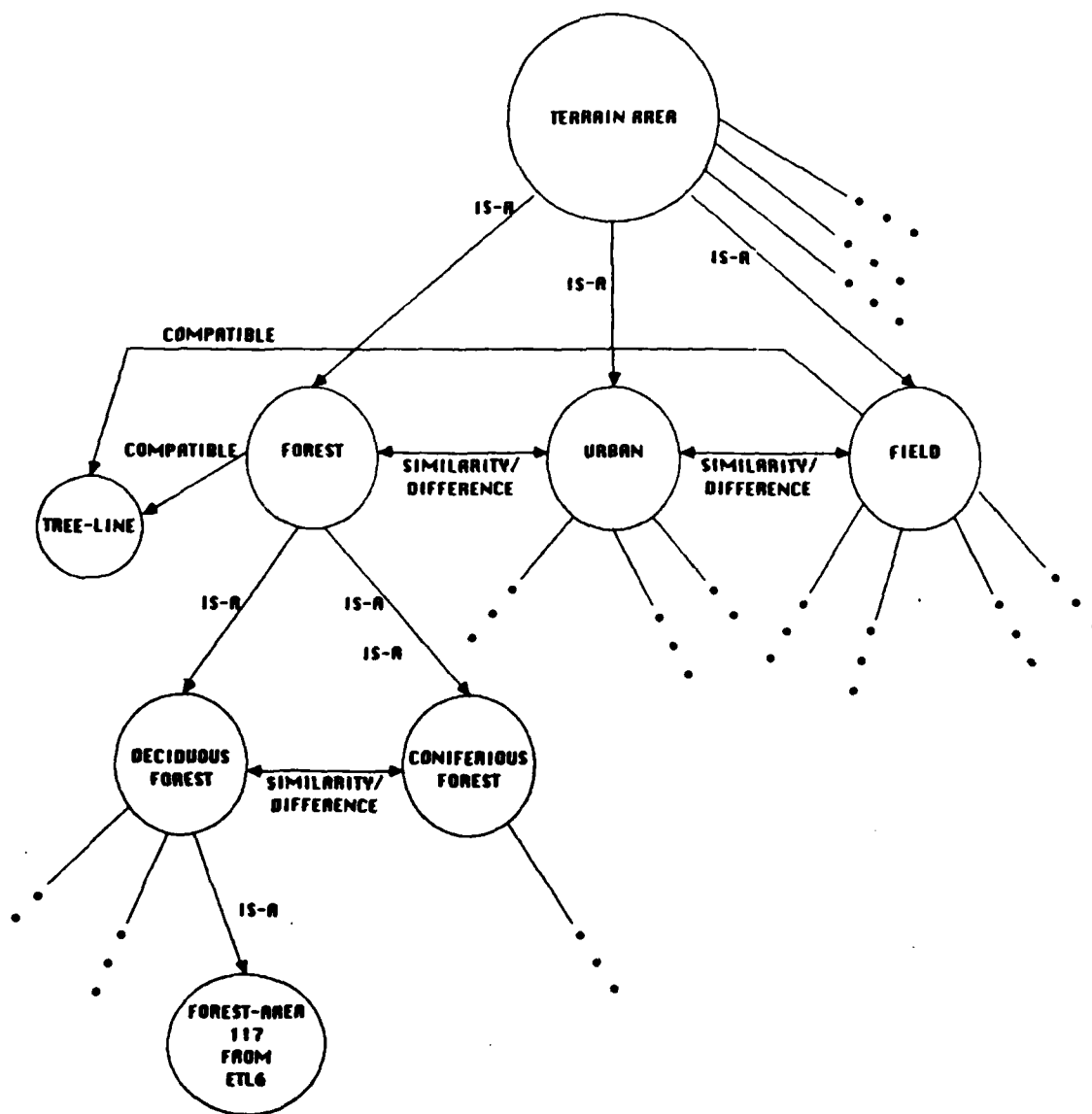
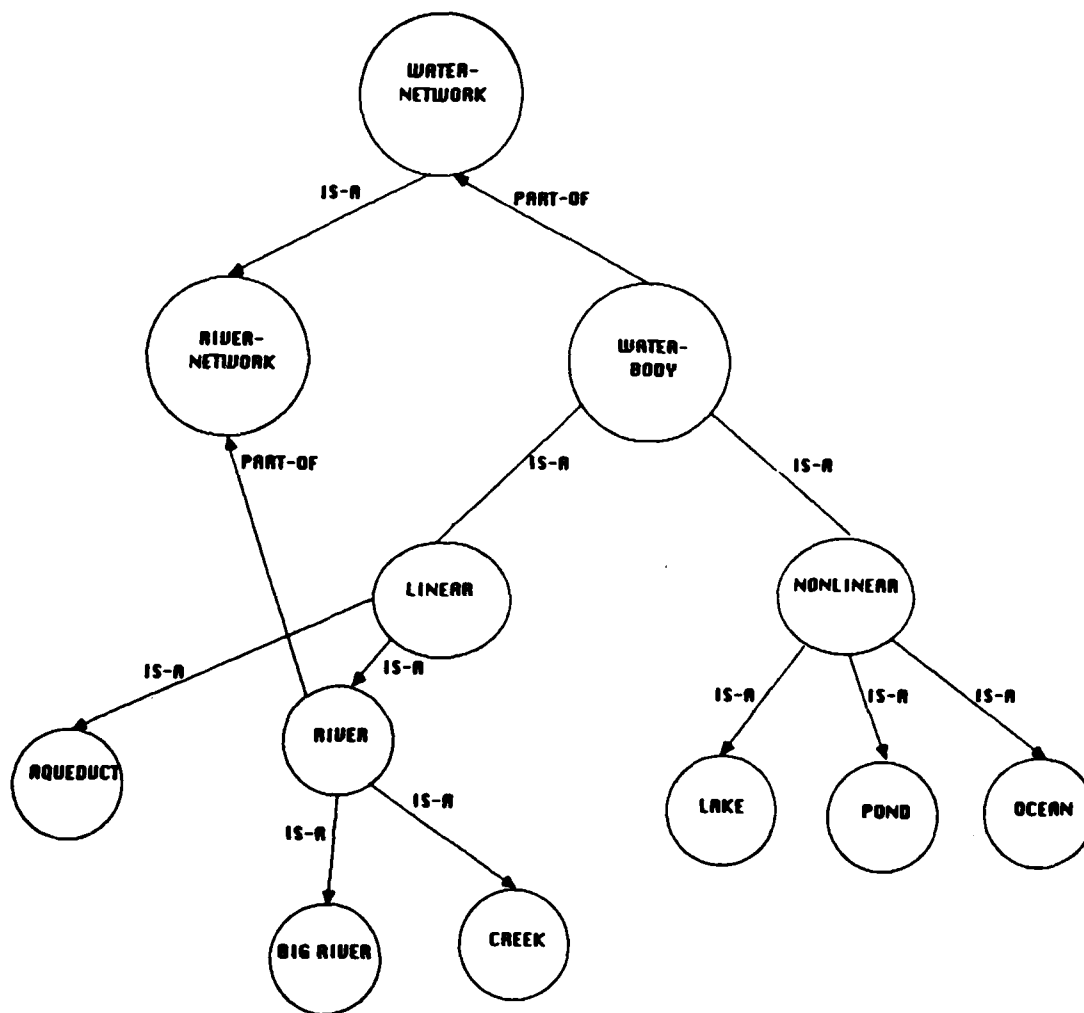


Figure 5-1: Object Subnetwork



**Figure 5-2: Object Subnetwork**

objects. There are two basic types: SPATIAL-COMPATIBILITY and SIMULTANEOUS-COMPATIBILITY. SPATIAL-COMPATIBILITY specifies allowable spatial relations between instances of objects such as intersection, alignment, contained-in, adjacent-to, and connected. SIMULTANEOUS-COMPATIBILITY specifies that a given image area can have multiple object types associated with it, as in a road object being simultaneously-compatible with an urban terrain area. The compatibility links can specify other objects which must be present as in the ocean being compatible with a land mass if a shoreline can be found. Compatibility links are directed so that one object can be compatible with another, but not the opposite. This is so that the generation of predictions along compatibility links can go in one direction if so desired. This is so one type of object always implies another type of object, but the reverse is not always true. The finders associated with compatibility links allow for contextually specifying relations between objects.

Compatibility has an optional numerical range associated with it from -1 (incompatible) to 0 (independent occurrence of objects) to 1 (highly compatible).

**PART-OF:** specifies the relations between necessary components of an object.

Our representation is generally two dimensional with three dimensional features, like shadows, being parameterized with respect to object height to derive two dimensional image characteristics. Three dimensional information could be incorporated in two other ways with very different system requirements. In one form, we assume a relatively precise three-dimensional terrain model associated with the images that are being interpreted. It is then possible to synthetically generate expected image properties and match these against an image. In this case, even though the model is three dimensional, it leads directly to image specific relationships. The generation of the predicted image features is automatic given an adequate sensor model and requires no inference processing. In the other form of three dimensional world models, there is a general geometric description of world objects and no a prior information specific to the image being interpreted. In this case, the system generates interpretations by manipulating these abstract three-dimensional models. Work to date on this in computer vision has found this to be enormously difficult. Additionally, the results of this inference processing take the form of compiling from a three-dimensional object to two-dimensional procedures.

This general format of this representation is compatible with several different evidential accrual schemes involving semantic networks expressing object interrelations, such as the Bayesian based scheme found in PROSPECTOR [Duda - 78] and the Force Structure Analysis subsystem found in ADRIES [AI&DS - 84], or the Relaxation based approach over certainty values associated with objects found in the ALVEN system [Tsotsos - 80]. The major question concerning these techniques are whether they can converge to an effective solution when dealing with large numbers of interrelated instantiated hypotheses, and how a priori compatibilities are determined and numerically evaluated.

## 6. SUMMARY AND FUTURE PLANS

The goal of this effort has been to establish the feasibility of automatically extracting linear features from radar imagery. The work has focused primarily on defining a general system architecture and considering key capabilities and techniques within that framework. Numerous basic segmentation procedures were considered and evaluated (including both existing algorithms and new techniques developed under this contract). The results concretely show the ability to extract image features. An image structure data base was implemented to demonstrate the ability to work with and manipulate symbolic representations of image objects. These capabilities were used interactively to determine the requirements of other parts of the system.

The results of this effort have been largely encouraging. The overall system concept appears to be robust and to provide the required capabilities. A sufficiently rich set of techniques were identified that perform well on SAR imagery to support automated analysis.

A full implementation of an automated Linear Feature Extraction System is planned as part of a Phase II effort in the Small Business Innovative Research (SBIR) program. That implementation will present the concept of a SAR Feature Interpretation Workstation. The workstation will support three basic uses. The first is for the interactive exploration or processing of an image. The second is for the online development of processing algorithms. The third is to interactively develop an autonomous vision system by generating new rules and editing the world object representation.

The future system will continue to be implemented in a LISP machine environment. It will potentially utilize an existing expert system framework and development tool such as MRS [MRS - 84], KEE [KEE - 85], or SCHEMER [SCHEMER - 85] for implementing the rule-based SAR Object Knowledge Sources and the Segmentation Knowledge Source.

The development effort will continue to be an evolutionary one. Representation will begin by implementing the declarative aspects of the object descriptions corresponding to feature vectors. This will be followed by the

implementation of Finders related in the network. Technique development will involve testing and adding new segments, edge finders, etc. to the system's range of capabilities and evaluating situations in which their use is most appropriate.

## **7. BIBLIOGRAPHIES**

[AI&DS - 84] "Advanced Digital Radar Imagery Exploitation Systems (ADRIES) Annual Technical Report", AI&DS TR-1040-01, April 1984.

[Agin - 73] G.J. Agin and T.O. Binford, "Computer description of curved objects", IJCSI, Vol 3, 1973.

[Asada - 84] H. Asada and M. Brady, "The Curvature Primal Sketch", IEEE Workshop on Computer Vision: Representation and Control, April - May, 1984.

[Ballard - 81] D.H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes", Pattern Recognition, Vol 13, No 2, 1981.

[Ballard - 82] D.H. Ballard and C.M. Brown, "Computer Vision" University of Rochester, New York, Prentice-Hall, 1982.

[Barrow - 78] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, and H.C. Wolf, "Parametric correspondence and chamfer matching: two new techniques for image matching", Proc., DARPA IU Workshop, May 1978.

[Blum - 67] H. Blum, "A Transformation for Extracting New Descriptors of Shape", Models for the Perception of Speech and Visual Form, MIT Press, Cambridge, Massachusetts, 1967.

[Burns - 84] J.B. Bruns, A.R. Hanson and E.M. Riseman, "Extracting Straight Lines", Proceedings: Image Understanding Workshop, University of Massachusetts, October 1984.

[Burt - 81] P.J. Burt, "Fast Filter Transforms for Image Processing", CGIP Vol 16, 1981.

[Burt - 82] P.J. Burt, "Pyramid-Based Extraction of Local Image Features with Applications to Motion and Texture Analysis", SPIE Conference on Robotics and Industrial Inspection, San Diego, 1982.

[Burt - 83a] P.J. Burt, "The Pyramid as A Structure for Efficient Computation", Multiresolution Image Processing and Analysis, (Rosenfeld Ed.), Springer Verlag, 1983.

[Canny - 83] J. Canny, "A Variational Approach to Edge Detection", AAAI-83, pp.54-58, August 1983.

[Duda - 78] R.O. Duda, P.E. Hart, N.J. Nilsson and G.L. Sutherland, "Semantic network representations in rule-based inference systems", In Waterman and Hayes-Roth, 1978.

[Genezereth - 80] M.R. Genesereth, R. Greiner, M.R. Grinberg and D.E. Smith, "The MRS Dictionary", Stanford University, HPP-80-24, December 1980.

[Goldberg - 84] A. Goldberg, "SmallTalk-80, The Interactive Programming Environment", Addison-Wesley Publishing Company, 1984.

[Grimson - 85] W.E.L. Grimson and E.C. Hildreth, "Comments on 'Digital Step Edges from Zero Crossings of Second Directional Derivatives'", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-7 No. 1, January 1985.

[Hanson - 78a] A.R. Hanson and E.M. Riseman (eds.), "Computer Vision Systems", Academic Press, New York, 1978.

[Hanson - 78b] A.R. Hanson and E.M. Riseman, "VISIONS: A Computer System for Interpreting Scenes", In "Computer Vision Systems", Academic Press, 1978.

[Hanson - 80] A.R. Hanson, E. Riseman and F. Glazer, "Edge Relaxation and Boundary Continuity", COINS Technical Report 80-11, University of Massachusetts, May 1980.

[Haralick - 81] R.M. Haralick, "The Digital Edge", In Proc. 1981 Conference Pattern Recognition Image Processing, IEEE Computer Society, New York, 1981.

[Haralick - 83] R.M. Haralick, L.T. Watson and T.J. Lafey, "The topographical primal sketch", In "Int'l Journal of Robotics Research", MIT Press (forthcoming), 1983.

[Hough - 62] P.V.C. Hough, "Method and means for recognizing complex patterns", U.S. Patent 3,069,654, 1962.

[Hummel - 83] R.A. Hummel and S.W. Zucker, "On the Foundations of Relaxation Labeling Processes", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 1-5, No 3, May 1983.

[KEE - 85] Kee User's Manual, Intellicorp, 1985.

[Kohler - 81] R.R. Kohler, "A Segmentation System Based on Thresholding", COINS TR in preparation, University of Massachusetts, 1984.

[Kohler - 83] R.R. Kohler, "Integrating Non-Semantic Knowledge into Image Segmentation Processes", Computer and Information Science, University of Massachusetts, September 1983.

[Krasner - 83] G. Krasner, "SmallTalk-80, Bits of History, Words of Advice", Addison-Wesley Publishing Company, 1983.

[Laffey - 82] T.J. Laffey, R.M. Haralick and L.T. Watson, "Topographic Classification of Digital Image Intensity Surfaces", IEEE Workshop on Computer Vision: Representation and Control, August 1982.

[Lawton - 85] D.T. Lawton and M. Callahan, "A Simple Tunable Interest Operator and Some Applications", Proceedings of SPIE, Applications of Artificial Intelligence II, Vol. 548, April 1985.

[MRS - 84] The MRS Dictionary, Stanford University, January 1984.

[Marr - 78] D. Marr, "Representing visual information", In A.R. Hanson and E.M. Riseman (Eds.), "Computer vision systems", New York: Academic Press, 1978.

[Marr - 80] D. Marr and E. Hildreth, "Theory of Edge Detection", In Proc. Royal Soc. London, Vol B, 1980.

[Marr - 82] D. Marr, "Vision", W.H. Freeman, San Francisco, 1982.

[Minsky - 75] M. Minsky, "A framework for representing knowledge", In P. Winston (Ed.), "The psychology of computer vision", New York: McGraw-Hill, 1975.

[Moon - 84] Moon, Stallman, Weinreb, "Lisp Machine Manual", 6th Edition, System Version 99, MIT AI Lab, June 1984.

[Nagao - 78] M. Nagao, T. Matsuyama and Y. Ikeda, "Region Extraction and Shape Analysis of Aerial Photographs", Proc 41CPR, 1978.

[Nagin - 79] P.A. Nagin, "Studies in Image Segmentation Algorithms based on Histogram Clustering and Relaxation", Computer and Information Science, University of Massachusetts, September 1979.

[Nevatia - 80] R. Nevatia and K.R. Babu, "Linear Feature Extraction and Description", Computer Graphics and Image Processing 13, July 1980.

[Nilsson - 80] N.J. Nilsson, "Principles of Artificial Intelligence", Palo Alto, CA: Tioga, 1980.

[Ohlander - 75] R. Ohlander, "Analysis of Natural Scenes", Ph.D. Thesis, Carnegie-Mellon University, April, 1975.

[Ohta - 80] Y. Ohta, "A Region-Oriented Image-Analysis System by Computer", Ph.D. Thesis, Kyoto University, Department of Information Science, Kyoto, Japan, 1980.

[Rosenfeld - 76] A. Rosenfeld and A.C. Kay, "Digital Picture Processing", Academic Press, New York, 1978.

[Rosenfeld - 76a] A. Rosenfeld, R.A. Hummel and S.W. Zucker, "Scene labeling by relaxation operators", IEEE Transaction Systems, Man, Cybernetics, Vol 6, No 6, June 1976.

[SCHEMER - 85] M. Cation, Schemer User's Guide, AI&DS, 1985.

[Tsotsos - 80] J.K. Tsotsos, "A Framework for Visual Motion Understanding", Technical Report CSRG-114, Dept. of Computer Science, University of Toronto, June 1980.

[Tsotsos - 84] J.K. Tsotsos, "The Scope of Motion Research: from Image Intensity Changes to Semantic Abstractions", Computer Graphics, Vol. 18, No. 1, Jan 1984.

**END**

**FILMED**

**12-85**

**DTIC**